DOCUMENT RESUME

ED 294 511 HE 021 560

TITLE Leveraging Information Technology. Track IV: Support Services.
INSTITUTION CAUSE, Boulder, Colo.
PUB DATE 88
NOTE 75p.; In: Leveraging Information Technology. Proceedings of the CAUSE National Conference (Tarpon Springs, FL, December 1-4, 1987); see HE 021 556.
AVAILABLE FROM CAUSE Exchange Library, 737 Twenty-Ninth Street, Boulder, CO 80303 (Individual papers available for cost of reproduction).
PUB TYPE Speeches/Conference Papers (150)

EDRS PRICE MF01/PC03 Plus Postage.
DESCRIPTORS Administrators; Automation; *College Administration; *Computer Software; Higher Education; *Information Technology; *Management Information Systems; Microcomputers; Office Management; Private Colleges; Productivity; Small Colleges; State Universities; Student Records; Technological Advancement; User Satisfaction (Information)
IDENTIFIERS Baldwin Wallace College OH; Barry University FL; California State University Fresno; California State University Hayward; *CAUSE National Conference; *Decision Support Systems; Eckerd College FL; George Mason University VA; University of Pennsylvania

ABSTRACT

Seven papers from the 1987 CAUSE conference's Track IV, Support Services, are presented. They include: "Application Development Center" (John F. Leydon); "College Information Management System: The Design and Implementation of a Completely Integrated Office Automation and Student Information System" (Karen L. Miselis); "Improving Managerial Productivity Using Microcomputers" (Mary G. Wilson); "How Do You Support a Campus-Wide Office Automation System and End-User System?" (Paul Tumolo and Steven Saltzberg); "Contingency Planning: A Call to Action" (Douglas E. Hurley); "Software Upgrades: Challenges and Solutions" (Susan A. Campbell); and "Implementation as an Ongoing Process, and Success as a Moving Target" (Louise S. Lee and Sharon R. Setterlind). (LB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED294511

Leveraging Information Technology

Proceedings of the 1987 CAUSE National Conference

TRACK IV: Support Services

December 1-4, 1987
Innisbrook Resort
Tarpon Springs, Florida

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

CAUSE

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

HE 021 560

Copyright© 1988 CAUSE

CAUSE, the Professional Association for Computing and Information Technology in Higher Education, helps colleges and universities strengthen and improve their computing, communications, and information services, both academic and administrative. The association also helps individual members develop as professionals in the field of higher education computing and information technology.

Formerly known as the College and University Systems Exchange, CAUSE was organized as a volunteer association in 1962 and incorporated in 1971 with twenty-five charter member institutions. In the same year the CAUSE National Office opened in Boulder, Colorado, with a professional staff to serve the membership. Today the association serves almost 2,000 individuals from 730 campuses representing nearly 500 colleges and universities, and 31 sustaining member companies.

CAUSE provides member institutions with many services to increase the effectiveness of their computing environments, including: the Administrative Systems Query (ASQ) Service, which provides to members information about typical computing practices among peer institutions from a data base of member institution profiles; the CAUSE Exchange Library, a clearinghouse for documents and systems descriptions made available by members through CAUSE; association publications, including a bi-monthly newsletter, *CAUSE Information*, the professional magazine, *CAUSE/EFFECT*, and monographs and professional papers; workshops and seminars; and the CAUSE National Conference.

We encourage you to use CAUSE to support your own efforts to strengthen your institution's management and educational capabilities through the effective use of computing and information technology.

INTRODUCTION

As professionals in an always-exciting field, we are constantly facing challenges to blend new information technologies into our institutions. It is important for higher education to develop environments that promote the use of information technology for strategic advantages, that allow faculty, staff, and students to benefit from existing technology, and that stimulate the discovery of new opportunities.

The 1987 CAUSE National Conference, with its theme "Leveraging Information Technology," offered the opportunity for us to share, exchange, and learn of new developments in information technology to improve and enhance our environments. The CAUSE87 program was designed to allow the fullest possible discussion of issues related to these new developments. Seven concurrent tracks with 49 selected presentations covered important issues in general areas of policy and planning, management, organization, and support services, as well as in the specialized areas of communications, hardware/software strategies, and outstanding applications.

To expand opportunities for informal interaction, some changes were made in the program schedule. CAUSE Constituent Groups met the day before the conference, as they did in 1986, but were given opportunities to meet again during the conference. Current Issues Sessions were moved to Thursday afternoon to provide some flexibility with time, encourage interactive participation, and extend opportunities to continue discussions with colleagues. Vendor workshops were offered for the first time this year, the day before the conference. The Wednesday afternoon schedule accommodated continued vendor workshops, vendor suite exhibits, and concurrent vendor sessions.

David P. Roselle, President of the University of Kentucky, set the tone for CAUSE87 with a Wednesday morning opening presentation expressing his commitment to the value of information technology in higher education. John G. Kemeny, past president of Dartmouth College and currently Chairman of the Board of True BASIC, Inc., spoke during Thursday's luncheon of new developments in computing for classroom learning. The concluding general session, Friday's Current Issues Forum, offered an exchange of philosophies about making optimal use of technologies on our campuses.

We were extremely fortunate to be at Innisbrook, a resort with outstanding conference facilities and great natural beauty (and weather)—a real distillation of the best of Florida.

Almost 800 people attended CAUSE87. Many of them described the conference, in their evaluation forms, as stimulating, informative, and memorable. We hope this publication of the substance of CAUSE87 will be a continuing resource, both for conference-goers and for those who will be reading about the conference offerings for the first time.

Wayne Donald
CAUSE87 Chair

Table of Contents

TRACK IV: Support Services 239

Application Development Center 241
John F. Leydon

College Information Management System: The Design and Implementation of a Completely Integrated Office Automation and Student Information System 253
Karen L. Miselis

Improving Managerial Productivity Using Microcomputers 265
Mary G. Wilson

How Do You Support a Campus-Wide Office Automation System and End-User Services? 275
Paul Tumolo and Steven Saltzberg

Contingency Planning: A Call to Action 285
Douglas E. Hurley

Software Upgrades: Challenges and Solutions 293
Susan A. Campbell

Implementation as an Ongoing Process, and Success as a Moving Target 303
Louise S. Lee and Sharon R. Setterlind

Track IV
Support Services

Coordinator:
Frank Thomas
University of Akron

In order to take full advantage of information technology, support units must be in place at all levels—consulting, education, maintenance, data access, systems development, security. Papers in this track describe the ways in which many institutions are redefining responsibilities and creating new opportunities to meet support needs.

Top: Karen L. Miselis, University of Pennsylvania. Below: Mary Wilson, Baldwin-Wallace College

Steven Saltzberg, California State University/Fresno, and Paul Tumolo, California State University/Hayward

APPLICATION DEVELOPMENT CENTER

John F. Leydon
USNH Computer Services
University of New Hampshire
Durham, N.H. 03824

The key to a highly productive Systems Development department is the proper use of a powerful Fourth Generation Language (4GL). The best method to insure the proper use of a 4GL is the successful operation of an Application Development Center (ADC). This presentation will provide the methods necessary to justify, plan, organize and implement an ADC. Techniques will be discussed to insure the successful use of a 4GL under the auspices of a development center. The two concepts (ADC and 4GL) are intertwined and inseparable if it is desired to get the maximum efficiency of the human resources involved in systems development.

The Application Development Center

presented by
John F. Leydon
George Kaludis Associates
and the
University System of New Hampshire

The time has arrived for data processing professionals to apply the tools and techniques of systems automation to the processes that have been used to automate the work environment in other departments. In most organizations the least automated functions are the systems analysis, systems design, programming and documentation techniques used by systems development personnel. The past few years have produced a tremendous number of automated assistances to aid the process of systems development. The rise of new series of buzz-words has heralded the advent of many new products and concepts. Computer aided software engineering, applications generators, relational data bases, fourth generation languages and information resource management techniques all have a part to play in the process of automating the development process. The challenge today is to collect the separate technologies into one concentrated methodology. The application development center is the mechanism which can provide the cement to bond the loose fitting processes together and establish a wholistic approach to the automation of the systems development process.

To better understand the concept of an Application Development Center (ADC), it would be helpful to envision an information center for systems development professionals. MIS executives have accepted the premise that more work can be accomplished if a small core of systems professionals were utilized to assist and encourage the end user to develop computer systems. Productivity is enhanced in two ways: the end user can immediately translate his needs into computer applications without the assistance of an intermediary, and the systems professional can spend her time automating the applications that require more advanced talents and expertise. It is important that those same MIS executives accept the concept that a small group of experts can improve the productivity of systems personnel by concentrating on the practice of technology transfer to assist the mainstream developer of automated systems. The results can be spectacular; productivity improvements of 50% to 60% are common. Typically, an ADC should be staffed with approximately 10% of the total number of systems professionals. If this 10% can improve the performance of the remainder of the staff by 50%, it is clear that the net result is highly positive.

The concept of an ADC is not new, but it has not caught on in MIS circles. IBM Canada introduced the idea at about the same time the information center concept was sweeping the industry. The information center has fully infiltrated the conventional wisdom and is now beginning to lose favor. The objectives sought with the introduction of the information center have, for the most part, been realized. There is a far more knowledgeable user demanding services of a different nature from our data processing departments. It is advisable to continue to encourage and nurture this type of end user, but a far more lucrative venture awaits our

attention. The productivity of MIS developers has remained static for a long period of time. For MIS departments to be successful in the future, new systems will need to be introduced far more rapidly, and the architecture of the systems will need to be conducive to fast modification.

To achieve major improvements in productivity, the entire process of systems development must be automated. Currently, the approach is to mechanize separate units of the development process and assume that the total effort will be enhanced by incrementally improving the individual processes. This approach could be deceptive. Peter Drucker, a well known management scientist, has stated that "the only things that evolve by themselves in an organization are disorder, friction, and malperformance". The automation of systems development cannot be left to chance; the process must be well planned and implemented in a controlled fashion. It would be inconceivable to automate the individual processes of the financial structure of an organization without an overall review of the interactions of the various elements. The same care needs to be given to the application known as "new systems development". The application development center is the mechanism that should be utilized to introduce the wholistic approach to the development process.

ADC Mission

The mission of the ADC is to support the systems and programming staff with dedicated resources to optimize and accelerate development and maintenance of quality application systems. The MIS organization needs to view the development process as an application. The practices and techniques brought to bear on a standard application, such as student records or human resources, must also apply to the systems development application. A project team is formed and provided with a direction statement, a budget, executive support and a requirement to prepare a project plan.

The project life cycle is applicable to this effort, and therefore the team must define requirements, investigate alternative solutions, select a solution, install the product(s), identify modifications, implement the modifications, prepare a test case, present a prototype solution, train the user community, prepare documentation, install the first application, review the results, and prepare to support the application with modifications and enhancements for the life of the product.

The ADC project team must insure that its objectives are in synch with the organization it was formed to service. For example, the team cannot decide that the best solution for the automation of the development process encompasses an environment supported by a VAX/VMS cluster if the university has all its applications running on an IBM mainframe. A more realistic example involves an organization that has set a strategic direction whereby all new systems will be purchased packages from established software vendors. This environment does not diminish the need for an ADC, but the products and processes selected will need to be more in tune with modifying and enhancing third generation software applications.

As an aside, the end is in sight for packaged applications developed in third generation languages. It is quite conceivable that a major investment in such a package at this time could be a significant waste of money. This concept will be further developed in the latter portions of this paper.

As stated above, the ADC is an information center for systems development professionals. As such, it must consist of physical location and a dedicated staff with access to software tools and adequate hardware resources. The idea is to leverage critical skills to increase the performance of all. The MIS executive must face the tough choice of removing the best and most seasoned development staff. It will be a fruitless exercise to initiate an ADC with the least productive members of the MIS organization. The staff is charged with effecting change by emphasizing leadership, responsibility and accountability. The natural response of the general MIS organization will be to resist change. The process will be eased if the ADC staff is recognized and respected professionals who will assume a leadership role to coax their peers to accept the new methods. The type of individual who will thrive in the ADC environment is one who naturally accepts responsibility and will not be satisfied until real improvements in productivity are demonstrable. The responsibility for results is held by the ADC staff, not the systems developers.

ADC Responsibilities

The actual responsibilities of the ADC organization will be as varied as the number of MIS departments employing one. The following list is typical and should be considered as a base to expand or contract to meet the realities of a particular environment:

1. Provide guidance in all aspects of the systems development process.

2. Establish a mechanism to measure the efficiency of the process to demonstrate productivity improvements.

3. Insure the appropriate allocation of hardware and software resources.

4. Insure adequate service levels of development staff.

5. Investigate new methods, tools, and techniques.

6. Train and assist development staff in the use of ADC facilities.

7. Provide data administration and test data system support.

8. Coordinate ADC functions with all other DP areas.

To provide guidance in all aspects of the development process requires a wide range of expertise. This feature of the ADC reinforces the concept that staff must be gleaned from the most experienced MIS personnel. If the full measure of productivity gains are to be realized, all areas need to be improved. The use of a fourth generation language may impact the programming phase of a development effort and bypass any improvement in other areas. The ADC should respond to the entire process and provide assistance in project management, systems design, programming, testing, documentation, and maintenance. Providing improvements to the entire spectrum of the systems life cycle requires that the ADC introduce new technology to improve the tools and techniques and, in addition, stress the analytical aspects of process and methodology.

It would be impossible to determine the success of the ADC without an accurate mechanism to measure the performance of systems development and the productivity gained by the ADC techniques. This area of MIS professionalism is woefully lacking. It is doubtful that 5% of MIS organizations have the tools available to accurately measure productivity. This dilema must be solved to justify the existence of an ADC. Senior management will require tangible proof that the expense of an ADC organization is warranted. The old performance standards will not suffice; the ADC must introduce a mechanism to justify its existence or face elimination.

Previous attempts to measure effectiveness centered on one of two mechanisms:

a. Lines of code produced per unit of work effort.

b. Satisfaction of defined requirements on schedule and within budget.

Both measurements are fraught with inconsistencies and inaccuracies. In fact, lines of code is a moot measure now that non-procedural languages are the norm. Completion of a project on time is more a matter of creative estimating than effective work effort. A better yardstick must be invented, or productivity will never be measured accurately.

Productivity is measured as the result of dividing the work product by the work effort. An effective productivity measure would determine the productivity, demonstrate trends, promote actions to improve work efforts, and show the results of such actions. In addition, the measuring device should support the estimating process by identifying discrepancies. A tool developed by IBM, known as "function point analysis", has the necessary ingredients to be an effective measuring device. This subject could be covered in a future paper if there is sufficient interest.

The ADC must also insure the appropriate allocation of hardware and software resources. In other words, the ADC must get its share. A basic proposition of this paper is to equate the ADC with other major

applications, such as student records or human resources. It is very important that the staff of the ADC demand equivalent allocation of resources. It is absolutely essential that each MIS professional have a dedicated terminal and preferably a microcomputer with terminal capabilities. A very attractive environment would provide a stand alone mainframe for systems development to insure that conflicts did not occur with the operation of production systems. Lastly, the need for a 4th generation language is essential, otherwise revolutionary improvements in productivity will not be realized.

As part of providing adequate resources, the ADC must strive to provide optimum performance of the available services. Studies have shown that the productivity of systems developers is directly related to on-line response time. Optimum performance requires subsecond responses. The study revealed that drastic improvements occurred with minor improvements in response time.

The following results were demonstrated when terminal response was reduced from an average of 2.22 seconds to an average of .84 seconds:

a. A 60% increase in transaction rates.

b. Output improvement of 58%

c. A 130% decrease in the error rate.

d. A 37% reduction in overall project cost.

e. A 39% decrease in the elapsed time of the projects.

This phenomenon was not immediately understood by the researchers. An assumption is that the ergonomics associated with terminal response and attention spans of the human brain are optimized when the interval is less than one second.

A rapid turnaround of batch testing procedures is as important as terminal response. Fifteen minutes is suggested as a reasonable elapsed time from submission to completion. Terminal response and batch turnaround are considered to be attributes associated with productivity levels of systems developers. Assuming an accurate measurement tool is in use, the attributes can be varied to measure the impact on productivity. Vary the response time and measure the results. If the measurement tool is capable of tracking this type of variable, it can be considered an effective management tool.

The ADC holds responsibility for technology transfer to the remainder of the systems development staff. For the organization to be successful, the MIS executive must jealously guard the consultative role of the ADC staff. There will be tremendous pressure to include the experts in the actual development process. If the ADC is co-opted in this manner, it will cease to exist. The central idea is to pool the best talent and share it with the entire inventory of applications to be developed. Should this talent be utilized for development itself, the concept of sharing is lost.

Rather than identify ADC representatives as part of a project team, they should be cast in a liaison role. The role of the ADC liaison is to provide technical support in the use of the products and to enforce standards. This role (technology transfer) requires training, assistance, and constant investigation of new methods, tools, and techniques. New products are being introduced daily. To remain current, continuous investigation is a must.

The following has been included as part of the responsibilities of the ADC but is highly dependent on the structure of each organization. To provide the full range of assistance in systems development, the ADC can play a major role in data administration and test data support. Data administration is not to be confused with data base management. The data administration function is involved with the design and tuning of the logical view of data. The physical organization of the data and the navigation of the data base to retrieve data should be left to the technicians populating the DBMS world. The ADC staff plays the role of promoting the concept of data driven design through the use of a dynamic, interactive data dictionary. All future systems should not be viewed as a series of processes that collect, store, manipulate, and display data. Rather, the perspective must be that a single source of non-redundant, related data is available to support any number of processes. The staff of the ADC is available to provide the most effective view of the data to the end user to support a particular process.

The role of data administration can be extended to include a central support for all test data. From the dawn of time, MIS professionals have created test data for ad hoc purposes. With the completion of the project, the test data is released and lost forever. The next project team begins anew to create data for a specific purpose, and the next, and the next. The ADC can be used as a repository for all test data. New project teams can view the growing inventory of test cases to extract suitable data and deposit additional cases to the inventory. The most effective mechanism is to have the ADC as custodians of a series of routines that can instantly review the production master files, guided by a series of parameters provided by the project team. The test data is a microcosm of the live data base, tailored for specific functions. Effective regression testing can easily be performed using this procedure.

The final responsibility of the ADC staff involves coordinating the functions of the ADC with all other MIS groups. The following interactions will support the previously mentioned responsibilities:

a. Technical Services - Installing fourth generation products and performance tuning to insure optimum efficiency.

b. Computer Operations - Negotiating service level agreements and monitoring actual performance.

c. Systems Management - Coordinating the use of project control tracking data to integrate with the measurement of productivity trends.

d. Data Base Management - Orchestrating the effective integration of the logical data views with the physical structure of the data base.

The ADC should not have data security functions. MIS management must assign this responsibility elsewhere. If security is left to fend for itself, the ADC could be a vulnerable area for misuse of data. The ADC has the tools for rapid consolidation and dissemination of data and the data administration role. Together, these two assets could provide the base for significant security risks. It would be well advised to appoint a data security administrator who has no organizational bond to the ADC.

Functions and Organization

The extent to which functions can be supported within an ADC will be heavily dependent on the size of the organization it serves. The larger the organization the more intensive the support of the functions. But regardless of the size, the functions discussed in this presentation should be supported in some fashion. It must be understood that a number of functions can be performed by the same individual in a small MIS group. The size of an organization should not be used as an excuse to overlook any functions; smaller groups will only need to adjust the level and intensity of the service.

Two primary functions of the ADC must be given the highest attention. The reason for creating the ADC was to improve productivity. Thus, the main functions will be to pursue research and development of productivity tools and techniques and to consult and educate systems developers in the use of the products. The tasks associated with these functions are as follows:

a. Identify and maintain an awareness of the development problems to be solved.

b. Design selection criteria and evaluation techniques for new products and methodologies.

c. Identify new hardware, software, methods and techniques; evaluate and test those products.

d. Develop financial justification for selected products and pursue the approval process.

e. Produce and maintain product usage guides and introduce products for aiding ease of use.

f. Create an educational model and initiate education for the systems staff.

g. Define a pilot project, serve as an integral part of the project, and evaluate the results.

h. Above all else, promote a culture of high productivity.

To promote a culture of high productivity, the ADC staff will need to practice sales and marketing techniques. The professionals in systems development will not rush to the ADC for guidance and knowledge. The ADC staff must market its product to the systems staff and sell the concept of greater productivity. The active support of the senior MIS executive will be urgently needed in the early stages to ease and strengthen the sales potential of the ADC personnel.

The additional functions of the ADC are important but secondary to the two primary functions. Those remaining include process management, quality assurance, and data administration. Earlier, the proposition was made that the responsibility of data administration could be enlarged to include custodial responsibilities for all test data. To effectively perform the secondary functions, the tasks to be accomplished are:

Process Management

a. Perform a study, select and implement a systems development methodology.

b. Define and perform the function of Project Control Administration.

c. Integrate all existing productivity tools into the development methodology.

d. Install and operate an automated tracking and monitoring system for productivity measurement.

Quality Assurance

a. Review and improve quality standards to strive for perfection in the development of new systems.

b. Instruct in the use of and assure adherence to the quality practices of the development methodology.

c. Conduct project walk-throughs on request and emphasize the three major objectives of quality assurance. The system must be correct, reliable, and maintainable.

d. Periodically analyze project control reports to identify problems and take corrective action.

Data Administration

a. Promote the culture and understanding of data directed development.

b. Develop and maintain all common data definitions in a central dictionary.

c. Administer a central test data base for all applications.

d. Develop a mechanism to refresh the test data base with extracts of representative production data for all applications.

The functions of the ADC can be as many or as few as the individual organization can support. The limitations on the functions performed are primarily the result of the size of the systems and programming staff. The norm is to provide one ADC staff member for every ten MIS professionals. A single individual could perform all of the functions listed above, but it is doubtful that it would be a success. Adapt the functions to the size of the organization and do not risk failure by attempting too great a task.

An ADC will function in an MIS environment with ten or more professionals, but the chances of success are greater if the staff size is twenty and above. For organizations with staffs of less than ten, the management must provide the productivity tools and encourage the existence of an informal ADC. Open communication is an advantage of a small staff. Everyone is aware of the talents of their peers and the most talented are generally regarded as leaders. In addition, all are aware of the tasks that others are performing. In this environnment, close cooperation and open communication can substitute for a formal developmenmt center. The manager of systems and programming would assume many of the formal ADC functions as part of his/her job description.

The minimum staff should consist of a technical consultant and a technical analyst. The consultant should be selected based upon project management ability and interpersonal skills. The primary requisite for the analyst should be technical skills and peer recognition as a leader in the technical aspects of development work. The staffing norm is ten percent of the total MIS professionals, therefore there should be a consultant and an analyst for every twenty systems analysts/programmers. Ideally, the full complement of ADC staff would also include a manager, a data administrator, a project control specialist, a quality assurance analyst and clerical support. This level of staffing can only occur in the larger MIS environments, but it is important to reiterate that the functions can be performed in any environment. Do not use the lack of staffing as an excuse for not creating an ADC; the concept will work regardless of the size of the organization.

Justification and Summary

The tools available to systems developers will reduce by half the work effort to develop new systems. In addition, a system developed in a fourth generation language environment will require one tenth the work effort to maintain as a system developed with a third generation language (COBOL). The net effect on the amount of work than can be accomplished is dramatic. The improvements in programming maintainance will have the more significant long term impact. All software packages developed in languages such as COBOL will soon be obsolete. It is not cost effective to install a software package to save time and money if the end result is to expend ten times the required effort to maintain the system.

Currently, the amount of work expended on maintaining a system is sixty percent of the total effort. If the maintainance effort can be reduced by ninety percent, the savings generated by installing a software package is lost. It would be more economical to use a fourth generation language to develop the system internally or, better yet, buy a fourth generation software package. The net result will be to double the amount of output created by the systems and programming staff. It is not enough to install fourth generation tools. A dramatic increase in productivity is dependent on the correct and consistent use of the new tools. The Application Development Center is the vehicle that will insure success and transport the MIS staff to a more productive future.

COLLEGE INFORMATION MANAGEMENT SYSTEM
THE DESIGN AND IMPLEMENTATION OF A COMPLETELY INTEGRATED OFFICE AUTOMATION AND STUDENT INFORMATION SYTEM

Karen L. Miselis
Vice Dean for Planning and Administrative Information Systems
University of Pennsylvania
Philadelphia, PA 19104-6377

The College Information Management System (CIMS) is a unique, comprehensive, integrated office automation and student data system, operating in a dual environment with an office minicomputer and the University's central administrative computer. CIMS has revitalized the College advising process and allowed for much more efficient and effective tracking of over 6200 students. CIMS has benefited College students by providing advisors with more timely and accurate information, by supporting better tracking of student progress, and by providing extensive information on student behavior for the support of long range planning. It includes eight subsystems implemented in stages from June 1985 through Spring 1987.

COLLEGE INFORMATION MANAGEMENT SYSTEM

The College Information Management System (or CIMS) is a comprehensive, integrated office automation and student data system, which operates in a dual environment with an office minicomputer and the University's central administrative computer. To our knowledge, it is the only one of its kind with a number of unique characteristics. I would like first to present the context for the planning and development of CIMS and then describe the design and development process. I will then describe the system itself and how it works and conclude with a discussion of what we learned from the development of CIMS and what we plan for the future.

THE CONTEXT

Like many other large educational institutions, the University of Pennsylvania has centralized many of its student service functions and uses centralized student data systems to help manage them. Those systems were originally designed to support only central, operational needs in a batch environment, and they are now so archaic that they do not even do that very well. They certainly do not support the various University demands for strategic information for planning and decision making nor do they really support any of the individual school needs for operational or information support. It is also the case that each separate school has unique, local needs that could never be fulfilled by an exclusively central system.

The College of Arts and Sciences is the largest undergraduate unit at Penn, serving approximately 6200 students, with 500 standing faculty, 250 non-standing faculty and 400 graduate teaching assistants. While the Registrar actually processes registration and drop/add, the College Office advises the students on registration, collects and verifies forms and forwards them to the Registrar's Office, tracks students throughout their academic career, and certifies all College students for graduation. It also coordinates all of the advising and recording keeping efforts performed by the 28 academic departments and 14 academic programs in the School.

Considering the number of students being served, the College advising staff is quite small, with approximately 13 FTE professional/faculty staff and 15 clerical and support staff performing a wide variety of functions.

For several years the College Office has had to deal with a very difficult and sometimes impossible situation involving the management of information. For example, every year the College staff filed approximately 15,000 transcripts, processed 20,000 drop/add forms, oriented and advised 1400 freshmen, graduated 1400 seniors, sent and received 15,000 letters, and advised students in approximately 20,000 interviews. With so many students to serve in so many different ways with so small a staff, automated support was essential to the provision of quality service, but the central systems were not appropriate or useful. Thus, over the years since 1981, my office designed a number of small, automated systems to deal with individual tasks. While those systems worked well, by 1984 it was clear that they were quickly becoming small bandaids on a very large and fast-spreading wound.

In November 1984, I received approval from the Provost to develop a completely automated, integrated office automation and student data system. Provost Ehrlich was very concerned about "shrinking the psychological size of the University" and thought that this project might be a good pilot in attempting to do just that. We spent some time planning the project and the design and development process and began the project officially in January 1985.

PROJECT DESIGN AND DEVELOPMENT PROCESS

Expectations: We entered into this project with a number of expectations about its outcome and the changes which would take place as a result of its implementation. Our primary goal was to store all student information electronically. The practice in the College Office had been to store all information on a particular student in a manilla folder. That practice resulted in an enormous amount of filing, retrieving, and sometimes temporarily misplaced folders. If we could store all student information electronically, advisors and administrative staff would be able to reduce the amount of time wasted on handling folders and increase the amount of time available for quality service to students.

Since the two main functions of the College support staff were record keeping and word processing, we assumed that a good office automation system would greatly enhance the productivity of those staff members. We hoped that we might also be able to include direct update of data to the central systems, although we were pessimistic about the acceptance of such an idea by the other offices on campus.

Because of the older, centralized information systems, the method used to transmit information to those systems, and the limited, bi-weekly central updates to the system, academic advisors were often advising students with obsolete information. If we could include local update capabilities in our system, more timely information would always be available to advisors and staff alike.

Finally, we realized that there might be some resistance to the installation of a completely automated system not only because individuals might be afraid of using a computer, but also because the new system would require the redefinition of most jobs in the office. However, we hoped that with the proper training and support, the advantages of the new system would soon transform reluctance into excitement and satisfaction.

Happily, virtually all of our expectations were realized. In fact, we were able to include direct update to the central files in a number of areas. We realized this goal because the offices with whom we were working began to see how our project could reduce duplication of effort and in fact decrease their own workload.

CIMS Development Approach and Team Structure: At the outset of our project, we decided that we would not use the traditional system development approach to application design. Up until that time with any application, no matter who the user was, University Management Information

Systems (UMIS) performed the needs analysis, the functional and detailed design, produced the programming specifications, coded the system, tested it and produced documentation with limited user input and no user participation in the process. The inevitable result was a system that was not anywhere near as useful to the user as it could have been with the proper participation. The structure of our design approach and team was new certainly to Penn and had an enormous impact on the outcome of the project.

The CIMS team consisted of three different groups all working closely together: SAS Planning and Analysis, UMIS, and the College Office. I was the overall project director and was responsible for the coordination of effort for all three groups and all phases of the project. It was the first time that the leadership for any UMIS application development came from outside UMIS. There was also a UMIS project manager and a project coordinator for the College Office group.

My office performed the needs analysis and then worked together with UMIS on the functional design, which I then presented to the appropriate College Office staff members for comment and approval. The UMIS staff and my systems analyst then proceeded with the detailed design and programming specifications. After College approval of that document, UMIS programmers coded the system and performed some initial testing. My systems analyst and the College project coordinator then performed extensive testing, and, after problem corrections by UMIS, performed training and implementation. My systems analyst and the college coordinator composed all the user documentation and developed a new Office Procedures Manual to help the College staff with their new responsibilities.

In the initial design of CIMS, we were able to divide the system into subsystems and then proceed as described above for each of those subsystems. In that way, we were able to install the subsystems in a phased manner. The phased installation turned out to be a very good idea in a number of ways. First, it allowed us to see some results of our efforts relatively soon after the beginning of the project. Second, it allowed the College staff to ease into the use of the computer with the training taking place for each new subsystem helping to reinforce previous learning. About half of the subsystems were implemented during the first year of the project. During the second year, we implemented the second half and worked on fine tuning both sets of subsystems.

Despite complications and delays that frustrated all the members of the team, the result has been a very useful system. The staff have found it easy to use and helpful in carrying out their daily responsibilities. As time goes by, they find more and more ways to take advantage of its capabilities and are thus able to increase the amount and quality of service they offer. It is also the case that a number of the other schools at Penn have seen our system and requested it for their own use.

THE CIMS SYSTEM

CIMS consists of two major components, the office component and the data management component, which are completely integrated. The office component resides on an IBM System/36 minicomputer located in the College Office, using IBM software packages for word processing, electronic messaging and ad hoc query. Each staff member has a terminal or PC attached to the S/36 with a uniform menu offering them the option of working on the office or data management portion of the system. The System 36 acts as a 30-port 3270 control unit, thus giving each workstation high speed access to the University administrative computer. Since the S/36 is a relatively user-friendly machine, we were able to train an existing staff member with no computer experience to act as the System Operator.

The data management component resides on our mainframe IBM 3081 administrative computer. It is a collection of eight subsystems implemented in ADABAS, coded in Natural, and running under CICS. CIMS data, which are organized around student identification number, include both central University data and local College data. I would like to give you a general sense of each of these eight subsytems with a brief description of their special features.

The Student Information Subsystem was our first implementation in June 1985. In includes twelve screens displaying biodemographic and academic data on our students; such as admissions information, home and local address, parent and sibling information, varsity sports and financial aid information. Two of the special features are the "student snapshot screen" for advisors to get a quick sense of the student and his/her status just before an interview and the cumulative history screens which give the advisors a sense of the student's activity with the office. The history screens include correspondence history, a history of academic actions, a history of petitions and requests, and a history of academic probation.

The Interview Notes Subsystem, also installed in June 1985, consists of three screens and allows the advisors to take notes on the interview with the student and also to review past notes in reverse chronological order beginning with any date. The subsystem includes a checklist of keyword topics for quick notes as well as later analysis. One advantage of the topics checklist is that it gives advisors a chance to analyze their "collective experience" over time and to identify any chronic or emergent topics of concern among their student clients.

The Transcript Subsystem allows for online viewing of the transcript as well as enrollment in current courses. It allows the College staff to electronically update academic actions such as Dean's List or Leave of Absence or Study Abroad. Staff members can also print local, unoffical copies of the transcript. The transcript subsystem was one of the most complex and most useful in taking advantage of ADABAS to improve on the obsolescence of our current centralized transcript system. Thus we are able to make changes and make decisions based on those changes sometimes up to several weeks before those changes are recognized by the centralized system.

Two of the transcript screens fully reproduce the University's online transcript but with the added feature of a temporary grade column along side the official grade. The registrar does batch upload of grade changes from paper forms only twice a month. Thus to provide more timely information to advisors, College staff key in the grade change to the CIMS temporary grade column before sending the form to the Registrar. The temporary grade is then visible to advisors and initiates a recalculation of the relevent term and cumulative averages, which are then marked with an asterisk to show that they include temporary grades. When the grade becomes official, the temporary grade is cleared and the asterisk is eleminated from the averages. This temporary grade feature has revolutionized the process by which College staff clear 1200 seniors for graduation only one week after finals.

The Forms Processing Subsystem was also incredibly complex to design and implement but wonderfully useful in cutting down on duplication of effort. It allows the College staff to communicate electronically on eight forms what they used to have to do with fifteen forms filled out in quadruplicate and sent to three different offices. These include forms for academic actions, student petitions, application for major/minor, application for foreign study, application for submatriculation into Penn's graduate schools, and request for adjustment of the student's bill. Some of the forms serve to capture information for the College's local use only, but others, such as academic actions and major/minor application, actually transfer data electronically to the central files. The subsystem's most important feature is its suspense file for actions that will take place as of an effective date at some point in the future. Such a suspense file exists nowhere on our current University systems.

Each screen in Forms Processing has a common format for processing student forms. The top part identifies the student, the middle part describes the petition, and the final part reports the decision on that petition. Each screen makes use of required fields to ensure that all forms are filled out completely.

The Graduation Tracking Subsystem is enormously helpful in allowing the graduate auditor to correspond intensively with approximately 1400 seniors each year, to check their records thoroughly and to clear them for graduation only days after final grades are recorded. It has three data screens and 22 prewritten reports.

The screens act as an organizing checklist for tracking each student through graduation mileposts. These include completion of an application to graduate, a course requirements worksheet, certification of major requirements, and completion of outstanding courses.

The twenty-two reports are selected from a menu and run as needed throughout the six-month-long tracking process. The reports perform selections of basic tracking information keyed into the online screens. Particularly useful are those reports which identify students with problems relating to their graduation candidacy. These give the College graduation auditor time to intervene and have a student correct the deficiency.

One of the reports generates a final graduation list which is used as camera ready copy for printing the graduation program. The final report electronically updates the permanent record of each successful graduate - noting the date of graduation and the degree received with all relevent honors earned. Both of these processes had formerly been done manually.

The Correspondence Subsystem involves the real integration of the office systems on the System/36 with the data management system on the mainframe. It is designed to capture essential summary information about every piece of correspondence either received in the College Office or sent from the Office. It includes four screens which process all incoming and outgoing mail in the College.

Each day a staff member scans every letter received and enters onto the logging screen: the student i.d., information on TO whom the letter was sent, FROM whom it was sent, a KEYWORD summarizing the letter's content, and to whom the letter was ROUTED for handling and response. A similar logging screen captures this information for outgoing mail. This screen is a specially designed addition to the System/36 word processing package. We interrupt word processing at the end of every edit session and transfer the user to the Correspondence Sent log screen. This semi-automatic process ensures that the user does not have to remember to log outgoing correspondence. If the outgoing letter is a mass mailing, the logging program (written in S/36 COBOL) will create one log record for each name in the mass mail data file. The user, however, fills out the log screen only once. Information from both logging processes is combined and uploaded to the student's record on the mainframe each night.

The summary data are then available to staff members in the form of a Correspondence History Screen in the Student Information Subsystem. By scrolling this screen, a staff member can view summaries of all correspondence relating to a particular student sent or received over that student's entire career at Penn. Further, should the staff member wish to locate a specific letter, the summary screen indicates the location of all correspondence - whether paper or disk file versions.

The Calendar Subsystem is the only one not yet implemented in CIMS. We tried to use IBM Personal Manager on the mainframe with a custom written interface to CIMS so that appointment information could be recorded on each student's record. Personal Manager turned out to be less than ideal for the task. We now plan to use the System/36 Personal Services with a similar interface. That system will allow the appointment secretary to schedule individual and group appointments of advisors. This is currently being done manually on large columnar sheets. The automated calendar will also allow the appointment secretary to note whether or not a student actually came for the interview.

The Query Subsystem uses Information Builders, Inc. FOCUS 4GL and its menu-driven user-friendly front end, TABLETALK as a query tool to allow ad hoc query of the CIMS system by selected staff members of the College Office. These are implemented with IBI's FOCUS-ADABAS interface facility. TABLETALK requires no computer sophistication on the part of the user and

allows him/her to produce reports, either on screen or on the system printer.

One of our primary goals in the CIMS design was to make data easily accessible to all College staff members for the production of ad hoc as well as prewritten reports. Such a capability would allow the College staff to produce labels and at least simple reports without having to depend on the expertise of the Office of Planning. In addition, we hoped that once the staff started using CIMS to imput and review data on-line, they would begin to generate ideas of their own concerning the analytic use of those data. That phenomenon has occurred. Our biggest concern now is proper training and supervision. TABLETALK is a very powerful tool, but if the user is not clear on the data elements and logic involved, s/he could draw incorrect conclusions from his/her own reports. Thus we have tried to train the selected users very carefully in analytic techniques as well as in TABLETALK, and we continue to work with them as they produce and use their initial reports.

CIMS Highlights: I have already discussed the major features of CIMS: the on-line access to information formerly stored in only paper files, the combination of both local and central data in the system, the distributed update capability, and the powerful query tools. There are a few other very important points about the system which I should mention here. CIMS contains a context sensitive on-line help facility, an extensive security system and an extensive system of error reporting throughout the system in order to maintain the accuracy of data. We also designed a special table maintenance facility which allows users to control changes to the CIMS tables. In addition, the design and development of CIMS stimulated a clarification and codification of College office procedures as well as a redefinition of virtually every job in the office. That process has increased individual efficiency and effectiveness and allowed for a much easier transition when there is staff turnover in the office.

LESSONS LEARNED FROM CIMS

University "Firsts": This project represented for the University a whole new era in administrative computing - an era when the level of technology and the sophistication of the user demanded an entirely different system and a different kind of relationship between the system users and the central data processing unit. Thus, the most significant "first" of CIMS besides CIMS itself was the formation and constitution of the system development team.

Ours was the first system with a project director who was outside UMIS. Ours was the first system in which a group of sophisticated users participated so closely in the general and detailed design, the testing, the training and implementation, and development of user documentation. This is, I believe, the way of the future. Since the beginning of our project, the University has initiated the development of new central student information systems, personnel/payroll systems and facilities planning systems. For all those projects, the director is external to UMIS and the users are closely involved in the design. In analyzing the varying

success of those projects, it is clear that those most closely involving users are the more successful projects. Our project would never have succeeded if the team had been configured in any other way.

CIMS was also the first University application using ADABAS with virtually all the coding done in Natural. That meant that we all had to go through an intense learn process in the CIMS project but that we could also take advantage of the sophistication of ADABAS to improve the utility of our system.

Another feature of our team which is not obvious was my intimate knowledge of the operations of the College Office, for I had run the College Office before taking over planning and administrative information systems. The fact that I understood so well the functions to be performed as well as the capabilities of the system to perform them helped to move the project along despite the complexities.

Our system was the first and remains the only application which incorporates both the host system and a departmental system in a completely integrated office automation and data processing system. So far, we have seen only advantages to such a system. The System/36 works very well as an office system and communicates well with our IBM mainframe so that we have a fairly user-friendly environment which promises to get easier and easier as time goes on.

Our system was the first application with distributed update capabilities. All of the new systems being designed now will include those capabilities, and our experience in CIMS is helping us to shape our future systems. In fact, as we design and implement those new systems, we will have the opportunity to completely change the nature of our administrative structure so that our services are much more efficient and effective.

With so many firsts, it was inevitable that we would make some mistakes, and we did. The biggest mistake, I think, was to be unrealistic about the implementation dates. While there may have been some good excuses for being unrealistic, the results were very damaging to the project. Once the deadline was past, many of the members of the team began to feel very negative about CIMS in that it was a source of guilt, that it introduced some sense of failure, and that it was interfering with other projects that had been planned. Thus, I have learned to plan the time frame for any project realistically no matter what the political pressures.

CIMS Costs: The hardware and System/36 software as well as a small amount of site preparation cost about $225,000. Personnel costs were at least $175,000, but most of that was for University personnel. Thus the cost was in what else they might have done instead of CIMS. There is no question that the benefit was worth the cost to the School of Arts and Sciences. Indeed, the cost was probably worth the experience and the CIMS model to UMIS and to the University as a whole.

CIMS Benefits: There are many benefits to CIMS, and they exist on three different levels. There is no question that there is increased

quality in advising and record-keeping because more accurate and timely information is available to any College staff member at all times. Advisors can provide more quality information to their students and can spend more time advising and less time looking for the appropriate information. It is also the case that now that the College staff members - that is, the individuals responsible for the welfare of our students - enter the data directly into the system, there is a greater chance for accuracy in the records and the staff feel that they have more control over the University bureaucracy. These particular benefits are quite visible to students on a day-to-day basis. Early on in the installation process, one Assistant Dean who had remained quite sceptical, confessed that it was wonderful to be able to correct a student's record before his/her eyes.

Another great benefit of CIMS is not so readily visible to students, and yet is probably more important to them in the long run. That is the ability to perform much better tracking on students as they proceed through their program. Too often in the past, the limitations of manual or partially automated systems made it impossible to perform all the tracking necessary. Thus, we did what was most crucial but not many other things that were also extremely important. We checked as best we could on students who might be in academic difficulty, but we never had the time to send out fellowship, award or graduate study information to our best students. Now we will be able to do that and much, much more. In addition, while the College as a whole will perform the major tasks, any individual advisor will be able to perform an ad hoc query and send a letter or information to any group of students he or she likes. That Assistant Dean will use query, for example, to ask for all his/her students who have not come in for the past two months and are taking calculus, will send the names and addresses to the System/36, and then send them all individualized letters to come in for a visit. In addition, the notation of that letter will go on the student's electronic record.

The most significant benefit arising out of CIMS will never actually be evident to the students. That is that CIMS provides us with extensive and detailed information that we have desperately needed but never had before. We will use that information to provide high quality support for our short term decision making as well as our long range planning. The evidence of that benefit has already been made clear to the Undergraduate Committee on Education in its efforts to create a new distributional requirement system. Our Dean has found this information invaluable in his efforts to enhance the College experience at Penn.

THE FUTURE FOR CIMS

We have already extended the benefits of CIMS to our academic advisors in the residences and we plan to offer some of the CIMS services to the undergraduate advisors in each of the academic departments. As more and more advisors join the system on line, we have less and less need to send them paper. They will no longer need hard copies of the transcript. They will be able to approve a major application at the terminal on their desk. This system will help us to encourage increased faculty involvement in advising. In fact, we have seriously considered the idea of putting the

rules and regulations on line so that they can be easily consulted by any advisor.

The most important future development for CIMS is the implementation by the University of new student systems. The University has already installed a new financial aid package. We are now in the process of modifying and installing a ADABAS Student Records System (SRS) package by Information Associates. SRS will include on-line course inventory, course registration, and drop/add functions, and will be an enormous improvement over our current central systems. However, it does not include many of the most important features of CIMS. Thus, CIMS will be modified as the new system is installed and will continue to be an enhancement for many years to come, not only for the College but also for several of the other Schools at Penn.

Improving Managerial Productivity Using Microcomputers

Mary G. Wilson

Baldwin-Wallace College

Berea, Ohio

ABSTRACT

Prior to 1986 little had been done to apply computer technology to the improvement of managerial productivity. The development of an integrated information asset accessible from low-cost, desk-top microcomputers via user-friendly software encouraged the development of managerial support applications at Baldwin-Wallace College.

After a year and a half tangible results include dramatic increases in both short-term and long-term planning and increased integration of planning among administrative units. While use of microcomputer-based managerial support systems is not without problems and concerns, the early results have encouraged us to expand efforts in the coming year.

The Challenge

Rapid growth in the student body, expectations of declining enrollments during the 1990's, the instituting of a merit compensation system, and an impending accreditation visit were among the factors that served to stimulate an administrative desire to increase managerial productivity. The existence of an integrated, computer-based administrative records system, and the availability of inexpensive yet powerful desk-top computing coupled with user-friendly software seemed to provide the resources needed to help managers be more productive. The challenge to the Information and Computer Services department was to provide the leadership, training and support necessary to realize the potential of hardware, software, and managerial enthusiasm in terms of increased productivity.

Background

Baldwin-Wallace College is a private, co-educational, liberal arts college founded in 1845 and located fourteen miles south of Cleveland, Ohio, in the town of Berea. The College has a full-time undergraduate enrollment of approximately 2200, with 1350 part-time undergraduates and 650 graduate students. There are 135 full-time and 140 part-time faculty.

The college's computer facilities include two Prime 750 super-minicomputers networked for administrative processing, and a Prime 2655 used for academic processing. The two administrative machines host a data structure that, while organized along functional lines, allows for easy intra-office sharing of data. By the time the managerial productivity project was begun virtually all clerical record keeping functions had been implemented on the administrative network.

Prior to the management productivity project the administrative use of microcomputers was limited to two or three applications where software was purchased from third-party vendors for specific functions. The current mix of administrative microcomputers includes 18 IBM PC/XT's or PC/AT's and 15 of the newer IBM PS/2 model 30's. (While 33 microcomputers may seem to be a rather small number, remember that this represents a potential productivity enhancement for approximately two thirds of the management team of Baldwin-Wallace.)

The administrative computer staff is comprised of four COBOL programmers, four production assistants, two administrators, and a secretary. There are four student employees to support administrative work. The department is organized to provide maximum flexibility. Work is organized into team projects within an environment that is personally supportive and frequently provides opportunities for professional development.

An important contribution of the Information and Computer Services staff is expertise in managerial and analytical techniques. The Director of Information and Computer Services has an MBA degree; several programmers and production assistants have undergraduate degrees in Business or business-related fields. A key element in hiring decisions is the ability to teach and convey technical information to a non-technical person in a clear, non-threatening manner. As a result of hiring policies and a 95+% on-time request completion rate there exists excellent working relations between Information and Computer Services and the managers of client offices.

Many of our administrative managers have come from the faculty; few have had either an education in management or experience in the business world. The lack of knowledge in such topics as statistics, linear programming, regression analysis, and scenario modeling was a hurdle recognized at the start of the project. Fortunately, managers recognized these shortcomings and were, for the most part, eager to remedy the situation.

The Process

Laying the Groundwork

The work directly with managers began in 1986, but the roots of the managerial productivity project reach back to the early 1980's. Changes in technology and management were manifested in an integrated information structure, greater involvement of client offices in information planning, and improved working relations between the Computer Center and the rest of the College administration.

While the immediate goal of early changes was to improve service to students, the long range goal was to develop both the technical and the organizational climate in which Information and Computer Services could facilitate improved managerial productivity.

To the good fortune of Baldwin-Wallace, as we were developing the internal environment for change, the computer industry was developing powerful desk-top computers and user-friendly software.

Before working with individual managers six key elements of the project were identified: securing the support of the College's officers, hardware selection, software selection, training, data transfer, and continued long-term support of managers. Probably the easiest task was securing the enthusiastic support of the College's officers. Their support was reflected in commitments of funds for hardware and software, support of policy recommendations, and most importantly as role models. Several vice-presidents were among the first to schedule training for themselves and they then encouraged their immediate subordinates to take advantage of the opportunities to become more productive. Establishing policies for the standardization of hardware and software, the centralization of purchasing

authority, the necessity of documentation and backup procedures, and recognition that central processing and distributed management applications could, indeed must, coexist, created a firm foundation upon which the project could be built.

Decisions on hardware and software were predicated on support concerns. Baldwin-Wallace elected to purchase IBM microcomputers to maximize maintenance options and to simplify software support. (Experience had shown that "MS DOS compatible" often meant that substantial time was required to make software operational and that documentation failed to identify all variations that might be encountered on non-IBM hardware.) IBM simplified the decision to forsake the less expensive clones by offering an aggressive discounting program.

Software was needed to do word processing, spreadsheets, data storage and retrieval, graphics, statistical analysis and communications. With a host of software packages available to do each task the primary selection concern was which software was the simplest to use and what was the quality of available training materials. Initially we had hoped that a single vendor for each type of software would be identified, and ideally there would be an integrated software system which would do most, or all, of the necessary tasks. Due to the variety of needs and prior client experience levels it was necessary to provide a choice of word processing packages (WordMarc and PFS:Write) and a choice of communications products (LinkMarc and Kermit). Lotus 1-2-3 was adopted as the standard for spreadsheets and graphics. The Arthur Anderson video tapes have been extremely useful in the instruction of Lotus 1-2-3. All client developed database applications use PFS:File while some of the more complex applications developed by the Information and Computer Services staff use DBase III+.

Before the project moved to working with managers the College's officers affirmed a policy that managers would have inquiry access to all appropriate items of the centralized database but that it would not be acceptable for an office to attempt to use a microcomputer to replace centralized storage of information used by more than one functional area. The cost of these policies is to increase the work of extracting and transferring data; the benefits are that a consistent institutional picture is available to all managers and the gains in data integrity accruing from a centrally administered system are maintained.

Training

Far and away the largest task in the managerial productivity project was, and continues to be, training. We moved simultaneously on three fronts. First, we began to offer our own staff (i.e. Information and Computer Services) training in microcomputer software. Some of the staff had limited experience with microcomputers, some had none at all. Microcomputers like those that would be provided to managers were installed in every Computer Center office. Those software packages which would be used in the project were provided along with the hardware. Staff members

were encouraged to attend both credit-bearing classes offered at the College and outside seminars. A goal at Baldwin-Wallace is to provide each professional employee several hours per week to be used for professional development; during 1986 and 1987 the most common professional development activity in Information & Computer Services was familiarization with microcomputing. Several staff members were interested in having microcomputers in their homes; they were loaned machines and software for several months.

All members of the Information and Computer Services staff were included in the training. This was done in the belief that each staff member would be involved in the support effort. The philosophy of team projects requires that there be a general understanding of problems and solutions by all team members as well as detailed knowledge by specialists. Hiring requirements were expanded for most positions to include microcomputer experience.

Second, we began to train our staff in the simple hardware maintenance of microcomputers; that is, replacing boards, installing printers, etc. One of our staff became the chief problem solver and trainer. When a problem with a microcomputer occurred, that person would read manuals, experiment, make vendor contacts and then when the problem was solved would gather an impromptu "class" of staff and share the experience gained. The new "microcomputer specialist" had no prior training in microcomputers but had a desire to learn and the tenacity to stick with difficult problems.

Third, we began to install microcomputers in administrative departments. The first ones were installed on carts, so that they could be moved from office to office. The software provided at this stage consisted of some demonstrations that basically ran by themselves and a few very user-friendly applications packages. The intent at this point was not to do any content instruction but to establish acceptance of a new tool. (We wouldn't have been too surprised if a little game playing was done in the early days, nor would we have been unhappy.)

At this point we discovered that although most managers didn't have microcomputer skills, most departments had at least one employee who did. Virtually every department had employees who had ideas for computer use, usually simple list keeping or forms development. For these simple tasks PFS:Write and PFS:File were ideal; they required no formal support, the documentation was easy to read, work could easily be changed, and simple results were normally produced in the first hour or two. If there was a downside to this it was that clients developed the bad habit of "build a quick little application now, and worry about documentation later".

Once our staff training was underway and a few microcomputers were installed in administrative offices, the most sensitive part of the program was undertaken: the training of the college managers. Our experience was that the College vice-presidents and department managers were cautious of

the new equipment and sometimes were apprehensive about being "students" again. A particularly supportive environment was needed.

A program of training was begun with several important components. The training was done by the Director of Information and Computer Services, a peer of the managers involved, and highly respected by them for his expertise and teaching skills. The training was done in the Computer Center microcomputer lab during a break in the school year, so that no traditional students would be accidental observers. The microcomputer lab was closed to all others during these sessions. Groups were no larger than five and were extremely informal, gathering around a microcomputer or two for observation and some hands-on experience. Five sessions were held. The first three sessions were an introduction to LOTUS 1-2-3, with emphasis on the nature of a spreadsheet and its uses. The last two sessions were the actual development of a spreadsheet reflecting long term trends in faculty departmental workloads. Special emphasis was on the analysis of the spreadsheet involved, more than on the clerical entry of the data, since our objective was to demonstrate planning and analysis tools.

We learned several important lessons from the first set of training sessions. One, get managers to put their hands on the keyboard and type. Have them learn that it's okay to make mistakes: the computers will not blow up and no little sirens are going to be sounded. Two, the training worked best when managers were solving their problems, not ours. We had them bring a problem that they felt was amenable to solution with a spreadsheet. Some of the problems were not amenable to a spreadsheet type solution - this too is a valuable lesson to learn. Three, managers are handicapped in that they do not generally have good keyboarding skills. This is a hurdle to overcome. Those software features that reduce keyboarding will be received enthusiastically.

Once managers had learned to use Lotus 1-2-3 we were well on our way. Using spreadsheets as the organizing and computational tool, managers were able to begin building and refining models. This required the organizing and writing down of many of the inter-relationships which managers had learned through years of experience. The ability to test many scenarios and tune decision values to obtain optimal results was exciting; previously the computational or programming requirements to do such analysis were prohibitive.

The quick successes with spreadsheets provided the incentive to explore other software. Communications software provided an avenue for the transfer of data from the mainframes to individual microcomputers. Word processing was soon coupled with graphics to generate more readable reports.

Support Issues

Once the first groups of managers had completed their orientation to microcomputing it was necessary to develop a more extensive hardware resource. The movable computers were supplemented by the installation of systems in the managers' offices. Experience during the training sessions had shown that the less diskette handling the better, so systems were installed with a hard disk. Also, managers were much more interested in solving problems than in learning the intricacies of DOS. A menu system was installed which eliminated the manager's need to learn anything other than the applications they would use.

Extensive one-on-one support from the Computer Center staff was necessary during this time. When a newly computer literate manager needed help, our staff responded, spending whatever time was necessary to teach the skills required. It quickly became obvious that more long term support would be necessary than had originally been anticipated. We began to search for a full-time microcomputer expert for our staff.

To minimize the trauma that would surely result the first time a hard disk drive crashed, a two stage backup procedure was implemented. Clients were cautioned to do regular backups of the hard disks to diskettes. A second backup of hard disks is performed by Information & Computer Services on a weekly basis. A Bernoulli disk system has been placed on a cart and an interface card has been installed in each hard drive system. The responsibility for the weekly backup has been assigned to a student assistant.

Some of our managers became very comfortable with microcomputers, using them on a daily basis. Others quickly transferred the responsibility for microcomputer use to others in their departments. They continued to use the output from the microcomputers as the basis for analysis and planning, however.

Outcomes

A year and a half later, the preliminary results have been encouraging, but sometimes unexpected. More technical support from Computer Center staff has been needed than we had thought. Requests to transfer subsets of data from the centralized data base to a micro have been greater than we had planned. There is a constant stream of questions to be answered and software to be installed, as users become more excited about the possibilities for data analysis. Managers have responded to the support by the Computer Center with their own support for our budget and staffing requests.

The project initially was aimed at providing quantitative tools to managers; this has been accomplished. The work of managers is not, however,

limited to dealing with trend analysis, dealing with exceptional cases, and planning for the future; there is also a substantial component of mundane reporting and record keeping. By improving the managers' ability to deal efficiently with the mundane, more time has been made available to address planning issues.

Baldwin-Wallace is currently experiencing significant enrollment gains. Because this is expected to be a short-term situation, staff has not been added to respond. The productivity gains provided by microcomputers have allowed us to take full advantage of a short-term opportunity.

The most gratifying results have been a dramatic increase in both short range and long range planning, and the increased integration of planning among administrative units. Some specific results of this program:

LOTUS 1-2-3 was used to analyze results of changing the faculty workload, with emphasis on class size and cost to the institution.

Academic department requests for new faculty are now evaluated in terms of five year trends in class enrollment.

Upper level section planning now uses lower level enrollment analysis. Studies of the implications of requiring students to take more upper level classes are being conducted.

The cost of moving the college up in the AAUP salary rankings has been analyzed.

Audit schedules for all financial departments are now produced on microcomputers, reducing the time required for the annual audit.

We are currently involved in re-accreditation studies. Most reports and analysis for this study are being prepared on microcomputers.

Analysis of capitalization strategies for construction of a new Health and Recreation Center was performed.

The Development Office tracks the return rate of mailing pieces to determine the effectiveness of segmented mailings.

The Registrar is using ACT computerized data to study trends in the geographic distribution of applications, and to shift emphases in recruitment.

Conclusion

Utilizing microcomputers has improved the managerial productivity at Baldwin-Wallace College. Improvements have resulted from reducing the time spent on mundane activities and by providing new, quantitative tools for planning. Managers are eager to expand their skills but require training to take advantage of both technology and techniques. A variety of training methods have been used, the most successful for us being the seminar approach in which the manager-student applied microcomputing resources to a current problem.

The early successes of the project have included specific, cost cutting applications and increased interest in planning. Those who were trained first have been missionaries to their colleagues, expanding the range of planning activities. The managers' desire to grow has exceeded the human resources of Information and Computer Services but the problem is a welcome one.

HOW DO YOU SUPPORT A CAMPUS-WIDE OFFICE AUTOMATION SYSTEM AND END USER SERVICES?

Paul Tumolo
California State University
Hayward, California

Steven Saltzberg
California State University, Fresno
Fresno, California

In recent years, two California State University campuses, Fresno and Hayward, have established "Information Centers" (IC) to support end-user computing, which includes campuswide office automation (O/A) projects. Over this time, the campuses have had to address such issues as: is the IC concept appropriate for end-user support, which services to provide, how to provide these services, whether to standardize, how to hire and keep good staff, and (of course) how to work with user steering committees. This paper discusses the experiences of these two campuses--what worked, what did not work, what we would do differently, and what we would do the same--in an attempt to enlighten the path for other campuses starting down the road to establishing end user support.

I. Descriptive Information

A. CSU, Hayward
The Hayward campus is an urban, commuter campus located in the San Francisco Bay area. The student population numbers approximately 9000 with 1200 faculty and staff. Office automation facilities include some 400 office workstations (IBM PC and AT&T 6300) linked via a fiber optic/twisted pair asynchronous network to a bank of nine AT&T 3B2/400 minicomputers which are in turn linked to the campus's administrative "mainframe", a CDC Cyber 830. Office automation is supported by an information center staffed by a manager (who also manages instructional computing support), a system administrator, two and one half consultants/trainers and four electronics technicians (who also maintain instructional equipment, other administrative equipment and the data networks on campus). The information center is a unit within Computing Services and works closely with other units in Computing Services to fulfill its mission.

B. CSU, Fresno
The Fresno campus is located in the middle of California's San Joacqin valley which is the major produce growing region in the state. The campus' student population is about 18,000 with about 800 faculty and an equal number of staff and administrators. The "office automation" solution for the administrative (including academic) offices are clustered microcomputers attached via a local area network. Specifically the microcomputers are manufactured by Convergent Technology, contain 80186 or 80286 cpus and are connected in clusters of six to twenty workstations that share software and hardware (hard disks, printers, and communication ports). Each cluster is in turn connected to the campus mainframe by emulating a 3270 bisynch cluster. The office automation and information center support is organizationally a part of the campus' computer center instructional area. The staff that specifically supports these functions consists of an assistant director of computing, a coordinator of office automation and training, several trainiers, two equipment repair technicians and half a dozen student assistants. The computer center operates three training/information center laboratories for the exclusive use of staff and faculty.

II. The Information Center

A. Concept
The goal of an information center is to promote and support end-user computing, thereby enhancing productivity and advancing overall computing efficiency at the university. An information center provides users with proper training, technical support, usable tools, data availability, and convenient access to systems, so that they may directly and rapidly satisfy a large portion of their information systems needs. Through training, user-friendly software, technical assistance, and consulting, an information center enables users to produce their own reports, efficiently communicate with other offices, access needed data, and develop applications specifications for use by Computing

Services analysts. As defined here, the university information center does not differ significantly from the corporate information center. However, there is one noteable difference. Where the corporate information center serves a more-or-less homogeneous population of administrators, engineers, or what have you, the university information center serves a truely dichotomous population of administrators and faculty. The unique goals, activities, work schedules, and expectations of the faculty create special challenges for the university information center.

B. Function

The primary function of the information center is to guide and support the integration of office computer systems (primarily microcomputer based systems) into the campus's overall communications and information systems. As such, the information center is an agent of change, facilitating the evolution of the university's information systems from yesterday's centralized batch systems to today's distributed online sytems.

III. Standards

A. The Case For Standards

It is difficult to imagine a successful information center without standards, that is, a list of products (hardware and software) for which the center offers support services. Among corporate information centers, the question of whether or not to have standards was settled in the affirmative years ago. Yet at many universities, the question persists. This is due mainly to the diversity of university faculty needs. Faculty's primary professional goals deal with instruction and research, not office products. Office products are simply productivity tools to most faculty. It is important to differentiate the need for standards in office products and and the need for state-of-the-art tools for creative instruction and research. The key to understanding the need for office product standards, however, lies in understanding the relationship of standards to the support function. The quality of support offered by an information center is dependant upon three factors: the resources available for support (primarily staff); the number of products supported; and the depth of support offered (how detailed are the support services, from merely listing products approved for purchase to offering training, consulting, installation, maintenance, etc.). The three factors, which make up the support "triad", interact to determine the quality of support offered. A change in any one factor, without compensating changes in the other two, will effect the overall quaility of support. In other words, if an information center's goal is to provide a resonable range of support services (i.e. a reasonable depth of support) on a large number of products, then the center would need a very large staff and budget. Most information centers have limited staff and budget, but still try to provide a broad range of support services. Hence the need for a limited number of supported products. The limited products that the information center

supports are the "standards".

B. The Standards Manual

The standards manual can prove to be an essential part of maintaining standards while minimizing political fallout. The manual aides in communicating to the user community both the need for standards and the services offered in support of the standards, as well as the standards themselves. The manual begind with a section on the concept of the information center, its goals and function. The manual should explain why standards are necessary and how they are set. It should list all supported products and describe the services offered in support of each product listed. It also contains a section on how the user acquires a product, how support services are obtained and a contact list for help or services. Finally, the manual must be widely distributed on campus and heavily promoted.

C. The Upgrade Challenge

Once it is agreed that standards are necessary and those standards are set, the challenge is not over. Change is the nature of our business. New products come on the market daily and new needs develop almost as frequently. Whether or not to add a product and when to change or upgrade a product become ongoing questions. Given the support "triad" (the three factors which interact to determine the quality of an information center's support), it is obvious that adding products to the support list without increasing staff or reducing services will degrade overall support. Hence, adding, changing, or even upgrading products must be seriously considered. (Note, for example, the upheavel being caused in some organizations by the appearence of IBM's PS/2.) One effective way to upgrade is to subsidize the change, i/.e. buy the user the new product on the condition that the user stop using the old product. If funding is unavailable for this, the old product can be de-emphasized (e.g. reduce the number of training session, newsletter articles, etc.) in favor of the new product. The goal of either of these two methods is to replace products rather than to add products. In either case, salesmanship is essential.

IV. Support Services

A. Consulting

The most effective consulting approach, but, alas, the most resource intensive, is the cradle-to-grave approach. Here, users are guided from the initial stages of planning for office automation onward. Support includes helping users determine which office functions should be automated, determining system specifications, acquiring and installing systems, and helping users write their own applications. User written applications should be developed in a language/data manager that is supported by the center so that the end user can receive some level of support. The information center can, when an application is nearly university-wide, choose to write and support certain applications. The center also assists offices in analyzing their

hardware and software needs, understanding specifications, and selecting appropriate tools. Supported hardware is inspected on arrival, tested, delivered and installed in the user office. Consulting and training on the use of the hardware is also provided. Software is installed for the user by center staff. An extensive software training series is provided, as are a hot line, drop in question answering, in-office consulting, and extensive "hand-holding". It should be clear that such an approach is very user contact intensive. True, it takes considerable resources, requiring one consultant and one repair technician per every one hundred office workstations to work well, but the results are impressive. Users acquire new skills rapidly and many offices are able to develop automated systems simultaneously. Users also feel that the information center is providing a valuable service which, together with the high visibility of center staff, facilitates acquiring additional resources for the center. When the resources are unavailable to properly staff an information center, an alternative approach could be the use of a department or school "guru". While the guru approach has some success in the instructional areas, it is less likely to succeed in the office products area. The designated guru is more likely to want to offer services to fellow academicians than to staff or administrators, and usually has a heavy teaching and research load. Another problem is that some areas will not have anyone available to serve this function. However, this approach is still feasible with adequate committment and, of course, some release time.

B. Training.

Training may take several forms, including classroom training, one-on-one sesions, self-paced training (CBT, video courses, etc.), and hiring outside trainers. The best approach combines each of the above, but with an emphasis on classroom training. Classroom training makes use of the expertise which exists in the consulting staff of the information center. It is an efficient way of training a large number of users on a given task in a reasonable amount of time. Self-paced training has not been as eagerly accepted (as classroom training) by most administrative staff, particularly clerical staff. However, self-paced training has a place in providing a means for review and in handling special needs, where the time and effort required to develop a classroom course for a limited number of users is not justified. Training for users with special needs can also be accomplished by using one-on-one training sessions, articles in newsletters, user group meetings or by sending the user to an off campus agency for training. Hiring an outside trainer is usually too expensive for the campus, given that training is an ongoing task with new staff joining the university on a continual basis. However, in some cases, outside trainers are a good solution. One such case is the installation of a new system, such as electronic mail, requiring many users to be trained in a short period of time or on a one shot basis.

C. Equipment Maintenance

A hotly contested issue is whether to maintain equipment in-house

or to contract for off campus service. If the campus already has an electronics shop which maintains other equipment, then it is usually more efficient and cost effective to expand this service to include office systems than it is to send repairs off campus. There are some exceptions, of course, most notable are hard disk repairs. However, even if repairs are done off campus, information center resources are needed to determine when repairs are needed, and to arrange and track repairs. In-house repairs allow service calls to be made in a timely fashion (often within an hour of the report) and when coupled with a "loaner" policy (where a failed device is replaced with a "loaner" should the failed device need to be removed from the office), promotes minimal down time and a very happy user community. The important issues that need to be faced when evaluating in-house or out-of-house repair are the true costs of both options. These costs include spares and parts, diagnostic equipment, office space and personnel costs, and (often overlooked) the cost of lost productivity when the office workstation goes down.

D. Centralized Data Bases

At the heart of administrative computing (with which office systems often interact) are the university's centralized data bases, such as student records and financial records. To be effective, office systems must be able to extract data from these data bases. Campus-wide standards, strickly adhered to, promote ease of interfacing office systems and central systems. Difficulties with communications connections and protocols are minimized and systems development and training are facilitated. With access to sensitive data, come issues of data security. Establishing clear procedures for gaining approval to access such data helps maintain the integrity of the data. However, once the data is downloaded to an office machine, only the office user can insure its security. Discussion topics at user group meetings and articles in newsletters help educate office users to the issues and problems involved.

E. Centralized Administrative Systems

The information center's function is to support end user computing. However, end user systems often interact with or otherwise depend on centralized administrative systems. Therefore, information center consultants need to be cognizant of administrative systems. They need to be able to advise users of optimal solutionsregardless of whether these solutions involve microcomputers or mainframes. They need to know how various products and existing resources can best be utilized and how new applications will effect old applications. They also need to know when to call in a systems analyst to make such determinations. Consequently, it helps to have both the information center and administrative systems in the same reporting line. Regular meetings between the two groups to discuss developments are essential, as are good person-to-person relationships between the groups. Rivalries must not be allowed to develop. It is important to promote the idea that the functions of the information center involve the efforts of both groups.

F. Information Services

A large part of the information center's job is to distribute information to users. Training sessions, of course, are one means of distributing information. There is also a need to distribute answers to common questions, helpful hints, product descriptions and the like. User group meetings are an excellent method for "getting the word out". Such groups should have regularly scheduled meeting, and be an informal affair perhaps with refreshments. The information center manager should encourage other unit managers to allow their employees to attend. Information center staff should serve as resource people to the user group, but the group should be under the control of the users themselves. Newsletters devoted to univeristy computing should contain regular sections dealing with office systems. It also helps to be creative in determining what these sections should contain. For example, instead of a dry, easily overlooked question and answer column, a "dear Abby" type column might be tried, where the questions are rephrased as amusing quandaries with equally amusing solutions. Or, establish a contest for the best user submitted tip of the month, or a crossword puzzle containing computer terms (an excellent way to promote computer literacy). Diversity and entertainment help relieve the feeling of being bombarded by overly technical material. Where long winded technical material (such as user guides) are necessary, they should be kept out of the newsletter. Instead, a series of technical bulletins should be established. If your office systems are interconnected, you can promote the use of the system by establishing a BBS (an electronic bulletin board) on your network. The BBS could contain such non-job related features as a jokes board or a recipe board. Although this may appear to "waste" resources, it is actually an effective way of getting office personnel to use the system and learn to feel comfortable and at home with the new technology.

V. Staffing

A. Obaining Adequate Numbers

Adequate staffing is critical to the support approach described above. The one hundred-to-one ratio was mentioned above. The ratio provides one consultant and one technician per one hundred office workstations. Such staffing represents a serious investment on the part of the university and the information center manager should continually lobby for adequate staffing. Promoting the center (see section VI) is an ongoing and important task which can yeild benefits when arguing for additional staff. A development plan for the center should be designed which includes a section on staffing. The plan must be realistic and well justified (particularly when calling for increased staffing) and have the full support of senior administrators. It helps to pass along to those administrators articles from the trade press which lend support to your plans. Keeping records of user contacts, training session attendees, project progress etc., and circulating reports based on such data also helps keep administrators informed of the center's needs.

B. Retaining Competent Staff

With many institutions establishing and expanding information centers, the opportunities for center personnel are currently quite good. Retaining good consultants and technicians is a problem which the information center manager must address (particularly at universities where salaries often lag behind industry averages). Listening to the concerns of your staff and acting accordingly is an effective strategy. Keeping the amount of time consultants and technicians spend on clerical tasks to a minimum is also a good recommendation (assign such tasks to clerical staff). Consultants are very "people oriented" and don't like to confront users with limiting policies and bureaucratic red tape. Consultants should be encouraged to refer all such questions to the information center manager. Another useful technique is to allow time in staff schedules for study, reading and attendance at seminars which allow staff to stay current or improve their skills. The opportunity to increase one's knowledge by attending off campus training sessions is often a strong point in retaining staff. In order to complete these objectives it is important to establish an annual budget for specialized training for information center staff. While it is often tempting to take on tasks without the necessary resources, this is generally a poor idea for an overworked consultant is ripe for moving on. It helps to remember that consultants and technicians are the heart and soul of an information center.

C. Optimizing Staff Resources

Augmenting the professional staff with student assistants can help stretch staff resources. It should be remembered, however, that student assistants are not (yet) professionals. They are perhaps best used in situations requiring minimal training and minimal knowledge of administrative policies and practices. For example, students can be effectively used to staff an evaluation center (where faculty and staff come to try out products) and as assistants in training classes. Students can also be used to do clerically intensive tasks (e.g., graphing data on user contacts), thus offering relief for the consultants. With careful screening, it is possible to use students in more demanding situations, such as sceening hot line inquiries. Another technique for stretching resources is the use of staff interns. In this approach, personnel from other departments serve as interns in the information center. The interns work a few hours a month or semester (what ever their home department can live with) in the information center where they may teach classes or answer hot line inquiries involving products they have used and know well. They may even be used to visit departments facing problems or issues which the intern has already solved in the home department. The home department (in return for giving up a few hours of the staff member's time) gains a staff member with increased knowledge of a product (the kind of knowledge acquired through teaching a workshop). The interns get a break from their usual routines, a chance to improve their skills, and experience which may, on occasion, lead to a career change. Among the more "traditional" ways to stretch staff resources, Computer Assisted

Instruction holds much promise for reducing the traing load. However, it has yet to fulfill that promise at most institutions.

VI. Staying Alive

A. Marketing the Information Center

The information center is a new concept for many and seldom appears on a university's organization chart. It is often not clear to administrators what the center does, why it is needed and (most importantly) why it is so staff resource intensive. There is often a feeling that once the equipment for office automation has been bought, the problem has been solved. To combat such mis-impressions, the information center needs to be "marketed". The university community needs to be informed of the center's mission; its function, goals and operation. Newsletter articles on the center itself are useful. The information center manager can also use various committee meetings to promote the center. It is also possible to impose on those administrators who are aware of the center's benefits to speak with those administrators who are not yet aware of the center's value. An excellent technique is to select a department which has not benefited from the center's services for some special attention. A consultant can visit the department and, working with the department head, can often quickly find areas where the departments operation can be improved or facilitated by taking advantage of the information center.

B. User Steering Committees

User steering committees are often thought of as a nuisance. However, if properly constituted and managed, they can provide much useful information on user needs and opinions. They can also help promote the benefits of the information center. If possible, the information center manager should sit on the committee and help determine its members. Steering committees can become counter productive if an antagonistic relationship between the information center and the committee is allowed to develop. This can be avoided by selecting members who are knowledgable concerning the center and who represent a broad range of positions within the university. Keeping the committee focused on user related issues, rather than the day-to-day operation of the center, is also helpful.

C. Evaluation

In-house, self evaluation is useful in preventing the information center from stagnating and keeping plans in touch with changing user needs. The information center development plan and usage data should be reviewed periodically by the center's manager with the help of the consultants and head technician. When the review is done on a regular periodic basis, change in the center becomes a smooth, evolutionary process, rather than a stressful, chaotic one. This is good for both users and center staff.

D. Charge Back

Running the information center as a cost center is an attractive

idea. User are easily made aware of the value of the center's services. However, in a university settings, charging for information center services is probably not a good idea. The center exists to facilitate end user computing. Putting a charge on services places a barrier between the center and those it is trying to help. Also, charge back encourages users to seek alternative solutions and make decisions based on cost alone, often without knowledge of the very interrelated nature of office systems. Given the function of the center in guiding the intergration of office systems into campus-wide communication and information systems, and the importance of maintaining standards, such maverick solutions, while costing less in the short run, could cost dearly when everything must work together. At the very least, charging for services should not be undertaken lightly and each service offered should be evaluated separately.

VII. Recommendations

This paper summarizes the experience of two campuses of the California State University system in providing for end-user support, concentrating on what has worked well on these two campuses. Through the experience we learned a few lessons regarding the initial establishment of an end-user support system for campus-wide office automation. First, thorough needs assessment regarding campus office automation requirements must be accurately and throughly developed at an early stage in the planning for office automation. Second, the actual users of the system, secretaries, clerks, etc., as well as managers and administrators should be part of the team which develops the system's specifications. Third, when a campus-wide network is being considered, a good deal of attention must be focused on the many details involved in connecting the various workstations and systems which will become part of the network. The state of network technology does not permit these details to be overlooked, for it is only after properly considering all the details that an accurate assessment can be made of the resources required and a plan developed. Fourth, the support problem is greatly eased by a fully integrated system, one in which all software and hardware is designed to work together as a unit. Finally, and most importantly, the university's senior administration must be made fully aware of the scope of the challenge from the very earliest planning stages. It is particularly crucial that the administration understand the need for adequate support resources, primarily staff resources, and that it commit to providing those resources. The view that the process is primarily a hardware issue must be altered. By its nature, the support of end user computing is staff intensive. This is the point which must be clearly understood.

CONTINGENCY PLANNING: A CALL TO ACTION

Douglas E. Hurley
Director, University Computing Services
University of Kentucky
Lexington, Kentucky

ABSTRACT

Computing centers are becoming such critical resources to the daily operations of most institutions that many could not survive long witnout them. A little foresight can significantly reduce the vulnerability of computing facilities, while improving the ability of the institution to continue operations in the event of an interruption in computing services, and to reducing the duration of such outages and associated costs of recovery.

Contingency planning should address a broad range of issues including assessing risk exposure, identification of vital and critical needs, and alternatives available to support those vital and critical operations, alternative information processing facilities in the event of an outage, and recovery and restoration of local computing facilities.

Contingency planning must be incorporated into the systems development life cycle and the daily operations of both operational departments and computing centers. Contingency planning must be viewed as an institutional obligation - not as a computing center plea.

CONTINGENCY PLANNING: A CALL TO ACTION

Introduction

Most computing professionals understand that the operation of their data center is a critical resource of their organizations. They also realize that this concentration of resource is sensitive and vulnerable to a range of natural disasters, man-made disasters, and even to malicious attack. Still, evidence suggests that nationally fewer than one data center in four has a current, tested contingency plan. For these centers, their strategy for dealing with an interruption of services is total risk acceptance. This attitude stems in part from the perception that little else can be done, in part from the search for and absence of global, easy, or miraculous solutions, in part from a reluctance to involve senior managers and end users, and in part from a lack of understanding of how to arrive at an effective plan.

Generally, most of us think of disasters in terms of events characterized by low rates of occurrence, high levels of uncertainty, and devastating consequences. Most managers will not see a disaster of this magnitude in their careers. For example, since the birth of the data processing industry, catastrophes can probably be numbered in the low hundreds. When these are compared to the hundreds of thousands of "installation years", it becomes obvious that a given installation can expect to see a catastrophe or major disaster something less than once every one thousand years. Most businesses that have encountered such events have recovered; few have had more than a rudimentary plan.

On the other hand, data center operations are becoming so integral to the operation of colleges and universities that many could not continue normal operations long without it. A little foresight can significantly reduce the susceptibility of computing facilities to damage, improve the ability of the institution to survive outages, and reduce the duration of the outage and associated costs of recovery. It follows, then, that we as prudent managers should invest the necessary foresight and consider the broad range of issues associated with contingency planning efforts.

Planning Scope

It is important to note the term "contingency planning" rather than disaster recovery planning. Recovery planning is certainly an important component of the overall efforts, but is not inclusive of the range of activities embraced under the broader concept of contingency planning. Contingency planning recognizes that data center services can be negatively impacted by a myriad of occurrences up to and including disasters of monumental consequence; and that these impacts can vary in duration as well as impact levels. Contingency planning begins with an assessment of risk exposure and addresses alternatives to reduce the vulnerability of the data center. Next, the planning process should address the impact of the absence of data center services on the ability of the institution to conduct its affairs and provide expected services. The next step should consider the business and service needs of the end users of data center services and how those necessary and essential operations can

be accomplished in the event of a sudden loss of computing support. And finally, the planning process should address those emergency strategies essential to enable the data center to contain the damage resulting from a disaster, to provide backup (albeit limited) computing services, and to initiate a recovery sequence which ultimately results in full and permanent restoration of services and facilities.

Plan Objectives

The objectives of a contingency plan are to make sufficient agreed-upon preparations, and to design and have ready to implement an agreed-upon set of tested procedures for responding to an interruption of services in the data processing/information services area of responsibility. The purpose of these procedures is to minimize the effect of an interruption upon the missions and operations of the various units of the university. The emphasis should be on safeguarding the vital assets of the organization and ensuring the continued availability of those institutional services and functions that are determined to be critical and vital to the organization.

A contingency plan should cover all aspects of the total or partial cessation of operations in each computing facility. This type of planning includes procedures as well as equipment and personnel for both automated and manual recovery procedures, both at the end user offices and in the data center. It also includes migration to, and operation of back-up sites, should that prove necessary. The personnel responsibilities associated with the plan must be specified and well understood. Complete or partial "disaster drills" and other more controlled methods of testing must be a regular part of the process.

Reducing the likelihood of a disaster and minimizing the extent of destruction through security and detection measures such as fire alarm systems, water detectors, sign-on passwords, specially locked doors, and the like, can be employed to minimize the accidental or intentional disclosure, modification, or destruction of data, or the loss of the means of processing data. However, security measures can fail with damaging results to the data processing facilities and thus to the institution. Contingency plans should be designed to reduce both the likelihood of an interruption or loss of computing services as well as to minimize the consequences of the loss of any computing resources or capabilities. Contingency plans do not merely mean planned responses to major catastrophes, but are also intended to reduce the detrimental consequences of unexpected and undesirable events of almost any magnitude.

As a practical matter, the greatest probability is that damaging occurrences will be less than catastrophic, and they will be confined to a small area of a total facility. However, the size and scope of a disaster and its effect on the computing operations are often not directly related. For example, a relatively small fire in the computer communications area could be widely disruptive to a facility's operations, while the loss of a few terminals in a completely destroyed building could be recovered rapidly. Computing operations are so interconnected that contingency plans must address the entire range of activities and services in each facility.

Given the growing institutional dependence upon computing services to support

a broad range of instructional, research, and administrative needs, an institutional commitment to address contingency and recovery planning is essential. Recovery planning in support of computing resources must be an organizational commitment and not merely a computing center obligation. Both the academic and the business impact of the immediate absence of partial or total computing support must be recognized and understood in order to proceed with developing a plan for recovery of essential functions and services.

The Need

Every organization must look at the consequences of loss of their computing resources and consider their risk exposure. It is simply good business practice for an institution to examine the possibilities for disaster and to estimate its exposure in each area. Translated into a college or university environment, those essential and vital activities of the institution which must be restored to an acceptable level of service within prescribed time frames must be defined. The definition process must involve the end users of computing service who, in concert with representatives from the appropriate computing facilities, can specify those activities which are essential and vital. In general, for an activity to be essential and vital, its unavailability should result in:

1. The creation of a life-threatening situation; or
2. A significant loss of assets (Physical Plant) or financial resources (dollars) to the university; or
3. An inability to meet critical and essential commitments to students, faculty, staff, or state or federal agencies; or
4. The inability to meet contractual (legal) and/or service obligations.

Some examples of the potential impact of the loss of computing services at a college or university include:

1. Inability to meet internal and external reporting requirements.
2. Inability to provide research and instructional computing support.
3. Diminished ability to maintain personnel and associated payroll records.
4. Inability to provide academic and instructional support services in such diverse areas as admissions, student records, financial aid, and human resources.
5. Reduced ability to register students and collect fees.
6. Inability to support instructional efforts requiring computing facilities.
7. Inability to support alumni constituency activities as well as institutional development and fund raising activities.

8. Diminish the quality of patient or client care at affiliated hospitals and/or clinics.

9. Inability to process patient/client bills and statements, resulting in reduced cash flow in hospitals, clinics, or service clearing operations.

Commitment to Action

Contingency plans must address both the "day-to-day" hassles as well as the "once in a thousand year catastrophe", and should be firmly implanted into the standards and operating procedures of the data center. The plan should be firmly grounded in the operational procedures of the users of the center and reflect their involvement and commitment. Through the explicit support of our end users comes the essential expression of institutional commitment which is vital in ensuring a contingency plan remains viable and functional. Recovery of essential computing services must be an institutional concern, in addition to a computing center concern. Those organization and institutions that have grasped this essential concept have viable contingency plans; alternatively, those institutions which continue to view recovery planning as merely a technical exercise, seldom have current and usable recovery plans. For recovery planning to succeed it must have institutional commitment in the form of recognized (public) acceptance of the need for a plan as well as access to necessary corporate and institutional resources in the form of dollars and personnel.

Alternatively, a recovery plan which lacks the imperative of institutional commitment will typically be little more than a technical blueprint to re-establish some pre-determined hardware capability at an alternative location. At best, this approach offers the computing center director a false sense of security which will last until the first execution (because of either a routine test or an actual disaster) of the plan. This approach often results from either an institutional mandate (auditors, statewide directives, etc.) or from the "something-is-better-than-nothing" syndrome on the part of the computing center director. In either case, the lack of broad-based user involvement and commitment in the recovery planning effort ensures that the institution will remain vulnerable.

While most computing professionals recognize both the need for recovery plans and their responsibility to ensure such plans are developed, they often are unable to marshal the necessary institutional commitments. Frequently, the discussion of recovery plans is framed around technical issues such as size, capacity, telecommunications, etc. In this context the costs (new) for a recovery plan must compete with other computing costs (new and recurring) for operating hardware and software. The decision reverts back to the computing professional from senior management, with the admonition to "work it out" within the context of other computing priorities. The burden shifts to the computing manager to justify the recovery plan from a computing perspective, instead of to the users of computing services to justify from an institutional and operational perspective.

The rationale and justification for recovery plans must be premised upon the business case of the sudden interruption of computing and communications

services and the resultant impact upon institutional activities. Both institutional management and computing/data processing managers should recognize that establishing and maintaining a dynamic and functional recovery plan is a necessary, ancillary cost of doing business; and that these costs are outside of and in addition to the current operating expenses associated with computing services.

Development of the Plan

As noted earlier, recovery planning must address a continuum of eventualities. The plan, of necessity, should be subdivided into logical and related actions to recover operational activities. While cognizant of the potential impacts of major catastrophes, the plan must also address more limited aberrations in the availability of computing services.

Planning for recovery activities should be embedded into the routine operational standards and procedures both in the computing center and in user offices. Both computing staff and user personnel should consider recovery objectives during routine design reviews and throughout the systems development life cycle. Recovery objectives should be integrated with and consistent with operational objectives. The ability to conduct at least at a rudimentary level, essential business functions in a manual or semi-automated fashion should be continually reinforced to both users and technical staff. Consider for a moment the consequences of developing a highly automated, technically sophisticated computing service or application for an end user, which in the absence of computing services would render that office or user group incapable of conducting even basic business operations. In the event of a loss of computing services, the likely ensuing scenario results: "The computing center talked us into this application, never told us that <u>their</u> equipment might fail, and never told us how to get the information for us to get our job done if the system wasn't available...". Recovery and restoration of business activities must be embedded into the fabric of the systems development life cycle.

Strategies and Procedures

Recovery planning usually begins at very basic levels with routine procedures, facilities, and policies. Effective strategies for recovery are premised upon establishing standard procedures for early detection of problems, analysis of their nature, and containment of their potential impact. To minimize potentially catastrophic events, electro-mechanical mechanisms such as smoke detectors, motion detectors, and heat sensors should be in place, as appropriate. Evacuation plans, notification sequence of civil and campus authorities, and emergency shutdown procedures must be established and broadly understood. Problem identification and tracking procedures are important. Regardless of the magnitude of the potential problem, end users and computing staff must share responsibility for problem identification and analysis. In this context, reliable and accurate communications among computing staff and end users is critical. Communications must be clear, responsibilities unambigious, and action consistent to ensure the nature and scope of problems are accurately diagnosed, analyzed, and disseminated.

Immediately following upon problem identification and analysis is containment. Containment should not be confused with resolution. Containing, or isolating a problem involves separating cause and effect, and differentiating the root problems from symptoms. Containment also involves interrupting and diverting a sequence, that left unchecked, may escalate to more disastrous levels. Containing and isolating problem situations is the necessary precursor to problem resolution.

Determination of responsibilities and establishment of an appropriate action plan are also necessary to accomplish problem resolution. As the sequence of problem identification, analysis, containment, and communication unfolds, critical decision points emerge. Clearly understood decision-making responsibilities must be established consistent with organizational structures and priorities. Critical time should not be lost because of uncertainty over who has what authority for what action in a given situation.

Actual problem resolution is the sum of the preceding activities: identification and analysis, containment and isolation, decision points and decision responsibilities. Problem resolution is a combined activity representing the coalescence of technical support and user expertise and judgment. The actual resolution sequence and the particular actions and staff required will reflect the nature of the problem. The problem resolution process in a recovery plan should be situational and temporal, determined by the nature of the problem, rather than by a narrow, rigidly proscribed decision/responsibility matrix.

Implementing the Plan

If the processes leading to problem resolution are embedded in the systems development life cycle methodology of the organization, then the formal declaration and adoption of a disaster recovery plan is relatively straightforward. And perhaps more importantly, the plan can remain dynamic and vital as evolutionary changes occur to either the hardware or software environments. When effectively implemented, a formal disaster recovery plan is an extension and formalization of various existing operational procedures into a coherent problem resolution and service restoration process. Specific operational details, such as backup facilities (hotsites, shell sites, etc.) are merely technical issues driven by operational needs. Staged recovery options which consider for example the first 24-72 hours after a disaster, the next 48 hours, and beyond, can be evaluated in the context of the service and business obligations of the institution. In sum, the actual recovery plan is a compilation of existing procedures developed in recognition of the business needs of the organization, accompanied by a technical support strategy which stages recovery options consistent with service obligations. Most importantly, the plan becomes a corporate statement of commitment, rather than a computing center's plea for relief. The scope of a recovery plan, adoption of a recovery strategy, and associated cost is focused at the institutional level, rather than at the departmental level.

When approached in this fashion, recovery planning can be both manageable and achievable. The various alternatives, responsibilities, and actions necessary to deal with the range of potential interruptions to computing services can be clarified and understood by both user and technical management. The essential

nature of recovery planning is expanded beyond recovering the computing center to focus on recovering vital operational and business functions and minimizing the impact of the absence of computing services. While problems and disruptions of service should scarcely be called routine, the presence of a contingency planning and recovery methodology as outlined above can ensure that an appropriate response to contingencies of various extremes is part of the standard routine of the institution.

SOFTWARE UPGRADES: CHALLENGES AND SOLUTIONS

Susan A. Campbell

Coordinator of Education and Training Services

George Mason University

Fairfax, Virginia

ABSTRACT

This paper will present a step-by-step methodology for introducing software upgrades within organizations. It will examine ways of including users in the decision-making process to help gain their support. It will also give approaches to evaluating software on the needs of the organization. In addition, hidden problems and costs result from software upgrades will be identified.

Introduction

In today's market, software versions seem to change as often as car models. At a seminar given in Washington D.C. last year, a representative from a major software company said that when their developers finish a new version, they go on vacation for a couple of weeks and then return to begin work on the next version. The constant introduction of new products into the marketplace forces managers to decide whether the latest product is worth the cost and disruption that usually accompanies change. This paper will present a process to determine whether or not to upgrade. It will also provide methods to implement an upgrade. While the information presented focuses on organizations with centralized management of office automation, individual users faced with the upgrade challenge may also find it helpful.

Software companies offer upgrades in a number of different ways. If the upgrade is taking place because the original version had bugs in it, the replacement version generally costs the customer nothing. What companies often do, however, is correct the bugs and add new features. When the company publicizes the new product, they advertise the new features rather than the corrected bugs. In this case, the customer can expect to pay for the new product. Often, a special trade-up price is offered to users who own one of the previous versions.

Identifying Hidden Costs

The Requirements of Software Companies

The software companies have various administrative procedures that they require when upgrading. As these procedures affect the amount of staff time needed to upgrade and the number of upgrades purchased, they are important factors to consider. Companies may require that customers:

1. Register their current version before upgrading.
2. Return all or part of the entire package before upgrading.
3. Return the old package after receiving the upgrade.
4. Purchase by a certain date to obtain a special upgrade price.
5. Pay the full cost of a new package if they miss a final deadline.

It becomes fairly obvious that research of the upgrade requirements must take place soon after the upgrade is released. This way, a quick decision concerning upgrades can be made and unnecessary expense is avoided.

Current Hardware Configurations

There are other factors to consider which can affect the cost of an upgrade. Current hardware configurations must support the software upgrade. For example, if the new software requires more memory or graphics support to run and the configuration does not currently provide this, memory upgrades and graphics cards should be in the cost estimate. Do not overlook printers when evaluating the new software. Make sure that the software will support current applications with all printers. In addition, make sure the printers will support all new features included with the software.

The Discontinuing of Older Versions

Also take into account that software manufacturers typically discontinue distributing older software versions shortly after new versions are introduced. This practice presents consumers with the following choices:

1. Upgrade all current versions to the new version.
2. Upgrade packages only for users who require the new features.
3. Shuffle the packages throughout the organization to make sure each office is not using more than one version of the software for compatibility reasons (license agreements permitting of course).
4. Select another software package that will better meet the needs of organization and adopt it in addition to the current software.
5. Replace all the current software with another software package that will better meet the needs of the organization.

Each one of the options above involves cost factors that should be included in the upgrade plan.

Discounts for Volume Purchases

Software companies frequently offer discounts depending on the quantity of upgrades purchased. These prices, along with special trade-up prices, usually have a time limit, after which they are increased or discontinued. If the organization does not have an office that centrally manages the purchase and distribution of software, the purchasing office may agree to combine all requests for a particular software upgrade on the same purchase order to take advantage of the discount.

Support Agreements

Generally, the more convenient and flexible a support agreement, the more it will cost an organization. Because of the enormous number of telephone calls they are

receiving from users, some software companies have decided to charge for technical support they previously offered free to registered users. Some of them have both corporate and individual support agreements available for purchase. Following, are some of the features that may appear in the various agreements:

1. A limited number of telephone calls (not necessarily solutions) to a technical support department on a central telephone number where the customers are helped on a first-come, first-serve basis. The number called is not necessarily toll free and customers may have to wait several minutes before they are helped.

2. A special corporate line allowing quick access to an individual or group of individuals assigned to the customer's account. Customers may have either unlimited or limited telephone calls over a specified period of time.

3. On-line help where a technician will physically tie into the customer's computer via a modem and call up the customer's session on his/her computer to analyze the problem and provide a solution.

License Agreements

The number of software upgrades purchased based on the organization's needs and the license agreement restrictions will affect total cost. Some software companies have license agreements allowing the use of a software package on more than one personal computer as long as it is not used on more than one concurrently. This practice means that if there is not heavy usage of a particular software package, two or more users could legally share one package and and use it on personal computers at their individual locations. Other software companies, on the other hand, have license agreements requiring that a specific software package be used only on the personal computer to which it is registered.

Staff Time

Do not overlook staff time as a cost factor. The upgrade process can take a great deal of time depending on the number and sophistication of the users affected.

Consider that staff may become involved in the following ways:

1. Evaluating and testing software.

2. Attending meetings to update the advisory committee.

3. Surveying and interviewing users.

4. Collecting necessary information and/or materials to take advantage of volume purchases.

5. Documenting the new features and how to use them.

6. Distributing software and testing it on users' configurations.

7. Providing follow-up support after distributing the upgrades.

8. Providing training to users on both an individual and group basis.

Based on the responsibilities given above, it should not come as a surprise to learn that staff could assume both temporary and ongoing time commitments with the upgrade program.

Deciding Whether to Upgrade

A Case Study

In 1985, a new version of a standard word processing package used by George Mason University (located in Fairfax, Virginia) was introduced by the software manufacturer. George Mason University has an Information Center (IC) which is responsible for the centralized distribution of new software and software upgrades to administrative offices. The IC staff began to seriously consider this new version because it would immediately solve three problems that users were experiencing. First, the older version was slightly incompatible with a new IBM clone (costing considerably less than the IBM PC) that had been adopted as a standard. Second, the new version allowed users to share printers via a local area network (saving the cost of purchasing additional letter quality printers) while the old version caused problems. Third, the new version allowed users to select one letter out of a series of letters created with mail merge (merging a list of variables with a primary document) and print it separately. The old version, on the other hand, required that the entire batch of letters be printed out each time a correction was made--a very time-consuming process. The major thrust to automate the University's offices with personal computers began in 1984. Formal training classes for administrative staff on existing software and hardware standards were started in June of that year and continue through the present. By the end of 1985 all the administrative offices had at least one personal computer and were using word processing (approximately 200 packages in total). A major investment was made in both the training of staff and the purchase of software packages. For this reason, in 1985, when the new software version was released, the University was not yet ready to consider changing to another word processing package.

Shortly after the IC staff decided to investigate the new version further, the software manufacturer released a second upgrade. For purposes of comparison, the older upgrade will be called upgrade A and the newer upgrade, upgrade B. The IC staff evaluated both software packages and tested them on the University's standard

configurations. They discovered that upgrade B would require upgrading memory on all the standard computers on campus which at that time numbered over 400. However, despite this cost, they felt the users might find the new features helpful. Taking into account the lack of user sophistication and knowledge of features, the staff believed it was important to administer a survey which would gather information and expand the users' knowledge base. The survey briefly described the new features of both versions with examples of how they could apply them and the users were asked to rate how helpful each feature would be in their work. The survey was one page long and it was printed on an unusual color paper so that it would not get lost on users' desks. The instructions for its completion were simple and direct. The survey was sent to every administrative office on campus (approximately 130) and generated a response rate of over 70%. When the responses were compiled, it was discovered that users indicated a need for the features in upgrade A over a need for those in the upgrade B. The survey results and the results from the evaluation and testing of the software versions conducted by the IC staff were submitted to a group called the Standards Review Committee. This committee evaluates and recommends standards for the University's administrative staff. It is made up of faculty members, administrative staff, and technical support staff. Faculty members are included because the department secretaries support them using standard hardware and software. After a careful analysis of the information provided, the Committee recommended that all word processing packages be upgraded to upgrade A. The funds allocation to purchase the upgrades was then approved by the Vice President for Computer and Information Systems.

Testing

One area that needs further elaboration than is provided in the preceding case study is testing. Testing deserves a great deal of attention prior to distributing upgrades to avoid problems later. Testing is important for individual users considering upgrades as well as entire departments or organizations. Customers have more leverage with the software companies if they are negotiating for a large number of software packages. Frequently, the software companies are willing to send evaluation packages to companies considering the purchase of software upgrades. Sometimes they are even willing to send a representative from the company to demonstrate the new features or to answer questions concerning the new software. The latter often depends on the size of the company and the priorities they have in terms of the groups to which they market. Universities are not necessarily a top priority with all software and hardware companies.

When software companies develop new software they do not always design it to work easily with previous versions. A file created in one version will not necessarily work with another version. Sometimes the compatibility problem is so great that it is

impossible to view a file created with another version. Other times only certain features will not work correctly. Deciding to upgrade software is not the same as deciding to buy new software. Testing of both the old version and new version is required. Look for performance problems in both the old and new features. The new features may cause the old ones to behave differently. The differences can be slight and still take hours of staff time to correct after distributing the upgrades to users.

Memory requirements are another factor that can cause surprises. Sometimes, the memory requirements given by the software company will not allow use of the new features to their fullest capacity. Examine the documentation for installing the software carefully to determine whether there is more than one approach with varying memory requirements. Unfortunately, even after someone with a great deal of technical ability has tested a software program, there is still a chance of overlooking something. Identifying test sites within the organization can prevent the distribution of software before the discovery of needed modification or major problems. It is probably best to select those offices with users who are already very enthusiastic about using the features available in the upgrade. Otherwise, they may resist making a change before other users in the organization.

Working with an Advisory Committee

If an organization has a centralized office handling office automation, and it does not already have an advisory committee representative of the target population, it may want to form a group for the purpose of recommending software upgrades. In this case study, the evaluation and testing was done by the IC staff. However, in other instances at this university where a choice was made between upgrading or adding additional software standards, a special task force reporting to the Standards Review Committee was recruited to do the evaluating. Having an advisory committee accomplishes the following:

1. It recognizes users who have become competent in the area of office automation.
2. It allows users to become involved in the evaluation and selection of their own tools.
3. It gives a group of users a vested interest in the success of an upgrade program and can help lessen resistance to change by the community.
4. It provides assistance to the central office for office automation that may be overextending its resources with the continual addition of new technology.

Administering the Upgrade Program

Maintaining Software Distribution Records

The first step in implementing the upgrade program is to locate the software packages needing upgrades. Hopefully, distribution records are already available. If not, this is

a good time to start collecting this information. Use a database management software program to develop an application to track the upgrades as they are completed and to add to the permanent distribution records. The following information is a basis for this application:

- office name
- contact person's name
- serial number
- version number
- peripherals (e.g., printers, plotters, modems, etc.)
- enhancements (e.g., graphics cards, memory upgrades, etc.)
- date upgraded
- name of staff person upgrading
- problems encountered and solutions

In the "problems encountered and solutions" field, reference a text file stored in another location if more space is needed.

If the software registrations are not currently maintained, start doing so with the software upgrades. Software companies may require copies when purchasing a corporate support agreement, replacing damaged software, or upgrading again in the future. Keeping them on file can save time later.

When the staff distributes the upgrades, it is a good practice to ask the person accepting the upgrade to sign a form which includes an inventory of the package and says that the person signing verifies that all items were received. In addition, attach a copy of the license agreement and include a statement on the form indicating that the person signing it has read and understood the terms and conditions of the agreement. Maintaining these forms can prevent misunderstandings at a later date.

Installing the Software Upgrades

To reduce the amount of follow-up support needed after distributing the upgrades, it is best to install the software prior to distribution. Determine the best installation procedures with the various configurations in the organization and document them. Ask all the staff members working on installations to follow the same procedures. In some instances, the installation may have to be done in a user's office because of software or hardware restrictions.

Providing Training on the Upgrade

If the organization has a training program, adding instruction on the new software version is advisable. Options include incorporating the changes into currently existing classes or offering separate classes focusing only on the changes. How and when this is implemented depends on a number of factors. They include: staff and facility resources, the number of users receiving upgrades, the cost of training, and the degree of difference between the older and newer version.

Preparing Upgrade Information Packet

Distributing an information packet with the upgrade will help ease the transition. Begin with a letter from the central office's staff congratulating the users on the acquisition of the new software and briefly describing the new features. It is important that a telephone number that the user can call for follow-up support is included also. If appropriate, attach a sheet explaining any new error messages the users may encounter and what they mean. Document any substantial feature changes, providing step-by-step instructions for how to work with the new features if the software company's documentation is inadequate.

Distributing the Upgrades

Resistance to change is more likely to occur in this step of the upgrade process. It is not enough to deliver a software upgrade to an office if a smooth transition is desired. Remember, removing a tool the user is familiar and comfortable with and replacing it with the unknown creates stress. The following steps should ease the anxiety created by this exchange and help avoid pitfalls:

1. Schedule a visit during a slow work period in the user's office.

2. Ask the user to set aside some time to allow the review of new features and test the software with him/her.

3. Review the organization and content of the software company's documentation with the user. Also review the information packet with the user pointing out the follow-up support telephone number.

4. Test the user's configuration with the software to make sure that the software will run. Include the printing of a file in the test. The testing verifies that the installation procedures selected for this particular configuration was correct.

5. Correct any installation problems discovered before making the exchange. If the correction is minor, it can be made in the user's office. Otherwise, reschedule the meeting and make the corrections in the central office. It is important that this

process does not inconvenience the user in any way. This will help avoid creating negative feelings towards the change.

6. Allow time to discuss any concerns the user may have regarding the upgrade.

7. If there are any new features that a user will encounter immediately during the course of his or her work with which he or she is unfamiliar, show the user how to operate them.

Obtaining User Support

Two ways to obtain user support were discussed earlier: 1) involve users in the decision of whether or not to upgrade; and 2) distribute the software upgrades in a way that gives users a chance to learn about the new features and ask questions without disrupting their work. Another possibility is the inclusion of an article in an organization-wide newsletter recognizing the contribution that advisory committee members and survey respondents made to the upgrade program. In addition, offering a seminar demonstrating the new features and providing examples of how users can apply them is beneficial. George Mason University offers lunchtime seminars that normally draw a group from 20 to 40 users. The IC staff attempts to create an atmosphere that is relaxed and informal. Users are encouraged to ask questions throughout the demonstration and voice any problems or needs they might have.

Conclusion

Purchasing upgrades should be the result of careful planning using cost-benefit analysis. Purchasing the latest technology is not always necessary to a productive operation. There are many factors to consider before deciding to purchase a software upgrade. While many of them may seem obvious, they are so numerous that they are easy to overlook. Since many of the factors associated with upgrades are discovered through practical experience, it is very important to document them. There are no "upgrade" textbooks. This documentation is especially critical with the continuous introduction of new products by software companies. Sharing software upgrade experiences between organizations is a good way to avoid many of the costs and pitfalls involved in the upgrade process. Upgrading can be a costly investment. Organizations must be careful not to let themselves be driven by what is occurring in the market. New technology is not necessarily the best technology.

IMPLEMENTATION AS AN ONGOING PROCESS AND SUCCESS AS A MOVING TARGET

Louise S. Lee
Director, Administrative Data Center
Barry University
Miami, Florida

Sharon R. Setterlind
Director, Computer Center
Eckerd College
St. Petersburg, Florida

Implementation generally refers to the initial stage of putting a system in place. With recent developments in database software and the trend toward user independence, it makes sense to view implementation as an ongoing process of review, evaluation, and change. The authors are IS directors at different institutions, running different administrative software packages on the same database system. This paper compares and contrasts approaches to continuous implementation at their respective institutions. Among the topics discussed are the role of the IS director as a facilitator of change, the importance of the steering committee in providing top level support, internal and external users groups, database issues and policies.

The term implementation refers to the initial stage of putting a system in place. It is a very special and exciting time. People are geared up to make an extra effort, there are schedules to follow and milestones to mark progress. Implementation plans are often ambitious and don't account for all the factors that will cause delays. Expectations are high.

Eckerd College and Barry University are independent institutions which both happen to be located in Florida. Each school purchased a different vendor supplied administrative software package and went through a similar implementation. Eckerd College bought AIMS from Axxess in December of 1981, and Barry University installed Datatel's Colleague package in December of 1983. Both software products run on the Prime Information system.

In neither case was there a clear demarcation of the end of this initial implementation phase. What happened, actually was a gradual realization that the implementation was not only incomplete, but also an acknowledgement that things were not exactly as they should be.

At both institutions, we began to evaluate our achievements. We looked at the conversion schedules, and we saw that we had met our objectives in that we had gotten each office on-line. Overall we had been successful. Users now had control of their own data. They were operating on real time and using shared data. A lot of duplication of effort and drudgery had been eliminated. The new systems were much broader than the old batch systems and many new functions had been automated.

We began to realize that things were not really finished, although every office was on-line. The registrar's office was up but was not producing an on-line transcript. Development was using their module as little more than a glorified word processor; although they were processing donations, they still could not get an accurate picture of what the major benefactors had given. Everyone was beginning to wonder when things were going to slow down and get back to normal; after all, the implementation was over.

In retrospect, what happened at both institutions was the result of a combination of factors. First, there were numerous problems and delays in the various areas that resulted in things not being fully implemented. At both institutions we had to deal with mass computer illiteracy, and we encountered a great deal of resistance on the part of some department heads.

Second, the interaction between various offices added a new dimension to the on-line system. Offices that had previously worked separately and independently on the old batch system were now thrown together on an interactive system with

shared data. This necessitated system-wide policies and new forms of communication. Administrative offices that are in different divisions of the institution had to learn to function as a unit. Prior to the implementation there was an intellectual awareness that interaction would be important, but the ramifications were not fully understood.

The third and most important factor was that both institutions found themselves on a different level of computing, having gone from batch systems to on-line real time interactive database systems. Users were doing things they never dreamed of on the old batch system. Admissions, for example, had done little more on the old batch system than maintain a name and address file to produce mailing labels; within a short period of time on the database system, they were creating documents and merging data from files. Soon they had developed a full fledged mailing system. In effect, the definition of success had changed.

We realized that the implementation was not finished, but we had lost all our props --- the conversion schedule, the vendor training, the enthusiasm. We began to think that some sort of a re-implementation was in order.

Case History: Eckerd College

Implementation at Eckerd began with vendor support but when the vendor went bankrupt, the task fell on the shoulders of the Computer Center staff. The initial schedule for implementation was changed and valuable time was wasted because the Computer Center staff found themselves having to learn the modules on their own before installing them in the individual departments.

As we moved throughout the campus bringing the database on-line, we began to realize the need for more communication between the users and the Computer Center. Eckerd had an established steering committee, the Computer Policy Group, which was made up of the Vice President/Dean of Faculty, the Vice President of Finance, two appointed faculty members, two appointed students, and the Director of the Computer Center as chair. This committee functioned well when it came to policies but did not meet the communication needs of the database users. An internal users group was formed and anyone utilizing the database was invited to attend. This group met to hear the Computer Center make announcements and dictate rules. Because of the scarcity of decision makers in this group, it was difficult to get priorities in order, or to effect any system-wide policies such as standards for data entry.

In March of 1985, the Computer Policy Group designated a subcommittee to recommend a long range plan for phased computer growth at Eckerd. This subcommittee sent out a survey to ascertain perceived computing needs and a long range plan for computing was designed. Many of the defined computer needs of the campus dealt with the expansion of the administrative software package, and the five items of the long range plan

that were critical to this process were:

1. Word Processing
2. Departmental Budget Information
3. Student Information
4. Scheduling of Campus Resources
5. Directory of faculty, staff and student information.

In order to expand the software package to include solutions to the long range plan, the existing database needed some fine tuning. Many modules were not being utilized to their full potential and the users needed further training to produce the results the package was designed to achieve. One area lacking in development on the users' end as well as the training was the query language of the database system.

The first step toward implementing the long range plan was the formation of the database task force. This group was made up of department heads, the decision makers of their individual modules. This task force was intended to complement the internal users group, and its purpose was to form an organized approach to the collection, storage and retrieval of data necessary to produce needed information in a timely manner. We expected to achieve this by analyzing the information flow at Eckerd and how the software package would support it. In particular, we planned to identify various ways in which the software could be utilized to attain goals set forth in the long range computing plan.

The database task force met for the first time in the fall of 1986. It became clear that many of the department heads, the decision makers, had a good understanding of their own individual modules but were lacking knowledge of the system as a whole. To provide this group with an overview of the whole system, the Computer Center held a two day off campus seminar entitled, "Walking through the AIMS System with You." This seminar was an in-depth view of the flow of data through the database and the interaction of the data throughout all modules. This seminar achieved in many ways an interaction between departments that the Computer Center had been internally trying to accomplish for years. During the last two hours of this seminar, we drew up a list of committees to evaluate problems such as name and address standardization and prioritization of master files. Our most significant accomplishment was the designation of responsibilities or ownership of data in the system.

After the seminar, the Computer Center set up intensive all day seminars with the person or persons who worked with the software on a daily basis. The sessions were designed to start at the beginning of the particular module and work through it one menu item at a time. The question we focused on at the seminar was, "How are you using the system and how would you make it better?"

The Computer Center became aware that there was a need for a formal structured way to record requests for changes and designed modification forms to gather information from the module review. The form included the date submitted, the date received, the expected completion date, and a detailed analysis of the problem. Each form had to be signed by the user's department head.

Many of the modifications were fairly easy to do and were not too time consuming; others required further evaluation. It was an eye opener to see how many users were putting up with problems that could have been fixed with a few lines of code or, in some cases, an explanation of a particular feature of the software.

The Computer Center developed an internal database to record all the modification requests. Weekly reports were generated and distributed to the database task force for review. The Computer Center used this modification requests database as a tool in planning the allocation of resources needed to accomplish the modifications as well as a tool in deciding which departments required further training sessions.

Because of the complexity of the query language of the database, the Computer Center set up training sessions in the individual departments. This allowed us to train them on their own files and data elements rather that taking a generic approach. These sessions involved all users in the particular department, and that helped create a feeling of unity in each office.

During the initial implementation phase and throughout the years, the role of the Computer Center has gone from a dictatorship where we set all the rules to a democracy where the users play a major part in the decision making. The Computer Policy Group, the database task force, and the internal users group have come to view the Computer Center as a customer service center where support and training are the primary functions. Although these groups continue to look to the Computer Center for leadership in the field of computing, they have a sense of ownership of the database and they are taking a more active role in decision making.

Case History: Barry University

During the implementation at Barry, we relied heavily on our software vendor for formal training and consulting. We knew that we were dealing with a limited resource, so we used the vendor support wisely. Gradually our staff took on more and more of the training. By the time the vendor training ended, we had a good system of formal training in place. As our knowledge of the software increased, we realized that we no longer needed help from the vendor in working out procedures. We began to conceptualize internal consulting as a function of the Data Center.

While we were developing the idea of internal consulting, the

Vice President for Business Affairs called me in and told me that the university was getting complaints from vendors who were not being paid on time. Recognizing a good opportunity, we jumped in and did what we later called a "module review." We interviewed everyone in Purchasing and Accounts Payable and mapped out what they do on flowcharts. Then we met with the vice president and the department heads over those offices and made recommendations. We agreed on some changes and before long the vendor complaints ceased.

The problems we found in the purchasing/accounts payable area had virtually nothing to do with the computer; they were people problems. People were simply doing things the way they had always done them. On the old batch system, for example, the people who accepted goods had been taught to hold the purchase orders until there was a sizable stack. Continuing to do this on the real time system caused bottlenecks in the accounts payable office.

We also discovered that the people in these two offices did not understand that they could retrieve information they had put into the computer and use it to help them do their work. They were still relying on manual systems to match things up. They were doing their transaction processing on the computer, but the computer was not doing anything for them. A few simple working reports changed that.

The actual changes that resulted from the module review were relatively minor. The flow of the paperwork was improved, some unnecessary tasks were eliminated, and the Data Center designed a few additional reports and put them on a user menu. Most of the work went into the analysis. The results of the module review were long-lasting and well worth the effort. A serious problem was resolved, and everyone involved came out of it feeling good about what they had accomplished. The two offices learned how to get effective support from the Data Center. Since the module review, they have never ceased asking for and getting things from the Data Center, including an inventory application. Once they understood how helpful data retrieval is, they began to take training more seriously. Several people in Purchasing have learned to use the database query language.

That first module review was such a success that we decided to formalize it and make it part of our regular services. We created a Module Review and Evaluation Committee which changes composition according to the area under review. The group includes department heads and key personnel from the offices, Data Center staff, and the vice president of any division involved.

A module review can be initiated by anyone, but normally it is a vice president who does so. The structure of the review varies according to the type of functions that are being reviewed. A review might encompass one function, one or more offices, or an entire division of the university. The length depends on the scope and complexity, and we never undertake

more than one module review at a time. Some of the reviews we have done were registration, student addresses, and we are currently doing one for the Institutional Advancement division which includes desktop publishing.

The primary objective of the module review is to step back and take a fresh look at a particular area, to focus as much of our attention on it as we can, and to reevaluate how the system is being used in that area. In the Data Center, we tend to work with bits and pieces of the system. We are constantly besieged by users from various areas asking for additions and enhancements. Almost all of their demands are legitimate, and we are so busy juggling priorities that it is easy to overlook things that may be very important. The module review gives us the opportunity to take an in-depth look at one area and give it the kind of attention it deserves.

At the same time we asked for the creation of the module review committee, we asked for a steering committee. There was a great deal of executive participation during the purchase of the system, but it began to wane as the implementation got underway. The module review insures a certain kind of top level participation, but we also wanted their guidance on long range planning issues such as upgrades and future directions.

The steering committee is composed of the Vice President for Business Affairs, the Vice President and Associate Vice President for Academic Affairs, and the director of the Data Center. The group discusses policies and issues relating to administrative computing. The existence of the steering committee allows the Data Center and the administrative department heads to be proactive in administrative computing, while insuring that what we do is in line with the institutional objectives.

The Administrative Users Group began as a forum for the exchange of information that affected everyone on the system. The early meetings were little more that a series of announcements; they were characterized by enthusiasm, naivete, and physical discomfort. We still meet monthly, but we do it in nicer facilities. We serve coffee and bagels, and the users dish up cynicism. We, the Data Center staff, welcome their cynicism because it is born of experience and is an indication that the earlier naivete has been replaced by a certain level of sophistication concerning the system.

The internal users group was the subject of some reevaluation after the initial implementation. Announcements were supplanted by discussions. The focus turned more to allocation of system resources, testing and installation of new software releases, and refining of system-wide policies and procedures. It became more of a decision making group.

One of the first things we did with the users group after the implementation was devote a long series of monthly meetings to presentations by different offices about what they do. One of the things we were not completely successful with in the initial implementation was the interaction among the different

offices. There was a real need to have each office understand what every other office was doing, so that they could empathize with the needs of the other users when we were negotiating procedures.

During the implementation phase, the Data Center dictated many of the policies relating to interaction. The purchasing office wanted to enter vendor information in upper case, but we forced them to do it in lower case because the development office might want to send letters to the vendors. Our gestapo tactics met with some resistance, but they were necessary during that initial stage. We are finding users much more willing to cooperate now that they understand what the other offices do and why. We in the Data Center had a unique vantage point on the system. We saw what each administrative office did, so we understood many things that another office would not. By setting up the individual office presentations, we tried to share that view of the administration. Apparently it was successful, because we now find that the users have more empathy with each other. The less we dictate and the more the users make decisions, the more they own the system.

Our internal users group has always been open to all system users, but now it is primarily attended by department heads and contact persons, or those key personnel who have primary responsibility for the system in their office. We used three of the monthly meetings this summer to define the responsibilites of the contact persons. Next month, we are forgoing the regular meeting for a staff development workshop on shaping change. The evolution of the internal users groups seems to parallel that of the external groups in that we spend less time on technical issues and more on management concerns.

We did a considerable amount of training in the early meetings, but that became unnecessary as we continued to formalize our training. We offer classes and each one has a syllabus. We use our continuing education software with local files to register employees for the classes. We identified various skill levels and related them to classes. We meet periodically with department heads to discuss what training needs they have and target employees for the different levels. The registration allows us to produce progress reports for supervisors.

We are learning to view problems as opportunities for solutions. Defining our role within the institution has been an evolutionary process. We have legitimized our functions of training, support, and internal consulting. The module reviews and our general approach to problem solving have helped our users understand our role. The more they understand our function, the more effectively they use our services.

<u>Compare and contrast what happened at Eckerd and Barry</u>

There are differences between Eckerd and Barry. Barry is larger and has more non-traditional programs. Eckerd had an

in-house computer system and Barry was using an outside service bureau prior to purchasing the current systems. The organizational structure of each institution is somewhat different. Administrative computing reports to different vice presidents, and administrative and instructional computing are combined at Eckerd but separate at Barry. We grouped people differently and used different names for them; Eckerd has the AIMS Task Force and Barry, the Module Review committees. These differences, however, are superficial and relatively insignificant in comparison to the similarities in our philosophy and approach.

We have both acknowledged that the system is dynamic. We accept change as a constant that we must live with. There are internal factors that cause change, such as growth, the proliferation of non-traditional programs, the trend toward more effective management, and personnel turnover. There are external factors as well that insure constant change such as trends in financial aid and changing reporting requirements. There are hardware and software changes that come with great regularity. The product life cycle of equipment is shrinking, and our hardware vendors make it prohibitive to maintain old equipment. Both our hardware and software vendors are producing frequent releases of their software. The releases invariably contain features that we need and want. Even if they did not, the vendors force us to stay current by refusing to support older versions of the software. There are other technological changes that open up possibilities that would revolutionize the way we are doing things. Scanners and voice simulation are examples of that type of change.

The other primary factor in the similarity of our approach at the two different institutions is the recognition of the importance of the human component of the system. A system is comprised of hardware, software, and people. And the people element is by far the most important.

Change cannot come about without sponsorship, and we have insured executive involvement by the existence of our steering committees. Change will not be successful unless the people who are affected by it are in favor of it. We have tried to insure that by encouraging user involvement thru the various groups and by promoting user independence. At both institutions, support and training are primary functions of the computer center.

What we have attempted to do, in short, is promote attitudes that are conducive to change and develop a flexible structure that will accomodate constant reevaluation and change. We offer training classes and demonstrations, we advertise successes, we focus on solutions, we try to maintain enthusiasm among our users, and we are constantly developing new tools which make the users more independent.

There is no simple formula for success. The people component insures that the system will always have gray areas and fuzziness. But if people are willing to acknowledge change

as a constant and if there is a structure that will support the mechanisms of change, success becomes more likely. We have learned that we cannot effect change without both top level support and user willingness. People will continue to resist change, and we will never be able to sit back, relax and say that we are finished. Implementation is an ongoing process and as long as we view success as a moving target and continue to work at it, we will be successful.